

# Oracle<sup>®</sup> Application Server

JServlet Toolkit Reference

Release 4.0.8.1

September 1999

Part No. A73045-01

**ORACLE<sup>®</sup>**

---

Oracle Application Server Release 4.0.8.1 JServlet Toolkit Reference

Part No. A73045-01

Copyright © 1996, 1999, Oracle Corporation. All rights reserved.

Primary Author: Sanjay Singh

Contributing Authors: Kai Li

Contributors: Zhou Ye, Sumathi Gopalakrishnan, Ramani Jagedeba, Gerald Bortvedt, Jun Wang, Raghavan Dasarathy

The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the programs.

The programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these programs, no part of these programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle is a registered trademark, and the Oracle logo, NLS\*WorkBench, Pro\*COBOL, Pro\*FORTRAN, Pro\*Pascal, SQL\*Loader, SQL\*Module, SQL\*Net, SQL\*Plus, Oracle7, Oracle Server, Oracle Server Manager, Oracle Call Interface, Oracle7 Enterprise Backup Utility, Oracle TRACE, Oracle WebServer, Oracle Web Application Server, Oracle Application Server, Oracle Network Manager, Secure Network Services, Oracle Parallel Server, Advanced Replication Option, Oracle Data Query, Cooperative Server Technology, Oracle Toolkit, Oracle MultiProtocol Interchange, Oracle Names, Oracle Book, Pro\*C, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

---

## Contents at a Glance

- 1 oracle.html package
  - 2 oracle.OAS.servlet.http package
  - 3 oracle.owas.wrb.WRB package
  - 4 oracle.OAS.Services.ICX package
  - 5 oracle.lang package
  - 6 oracle.owas.nls package
  - 7 oracle.plsql package
  - 8 oracle.rdbms package
  - A Class Hierarchy
  - B Locations of JServlet Class Files
- Index



---

---

# Contents

<b>Preface</b> .....	xi
<b>1 oracle.html package</b>	
IAlign interface .....	1-6
IHtmlItem interface .....	1-7
ITableFrame interface.....	1-9
ITableRules interface.....	1-10
ITarget interface .....	1-11
IVAlign interface.....	1-12
Address class .....	1-13
Anchor class.....	1-15
Applet class.....	1-20
BaseFont class.....	1-28
BlockQuote class .....	1-31
CheckBox class .....	1-33
Color class .....	1-36
Comment class .....	1-40
CompoundItem class.....	1-43
Container class .....	1-49
DefinitionList class .....	1-55
DirectoryList class.....	1-59
DynamicTable class .....	1-61
Embed class .....	1-66
Font class.....	1-71
Form class .....	1-74

FormElement class .....	1-78
Frame class .....	1-86
Frameset class .....	1-91
HeadLink class .....	1-96
Heading class .....	1-99
Hidden class .....	1-103
HorizontalRule class .....	1-105
HtmlBody class .....	1-108
HtmlFile class .....	1-114
HtmlHead class .....	1-118
HtmlPage class .....	1-125
HtmlStream class .....	1-135
HtmlStringBuffer class .....	1-140
IHtmlItemImpl class .....	1-146
Image class .....	1-149
ImageMap class .....	1-154
ImageMapArea class .....	1-157
Item class .....	1-160
LineBreak class .....	1-177
Link class .....	1-179
List class .....	1-183
ListItem class .....	1-189
MenuList class .....	1-192
MetaInfo class .....	1-194
Option class .....	1-197
OrderedList class .....	1-200
Paragraph class .....	1-202
PasswordField class .....	1-205
Preformat class .....	1-207
Radio class .....	1-209
Reset class .....	1-212
Script class .....	1-215
Select class .....	1-219
SimpleItem class .....	1-223
Style class .....	1-227

StyleElement class.....	1-233
Submit class .....	1-238
Table class .....	1-241
TableCell class .....	1-250
TableDataCell class.....	1-253
TableHeaderCell class.....	1-260
TableRow class .....	1-262
TableRowCell class.....	1-268
TextArea class.....	1-271
TextField class .....	1-275
UnOrderedList class .....	1-277
XObject class .....	1-279
HtmlException class .....	1-291
HtmlRuntimeException class.....	1-292
HtmlInternalError class .....	1-293

## 2 oracle.OAS.servlet.http package

HttpSession class.....	2-4
InactiveSessionException class .....	2-11
InvalidNameException class.....	2-12
NullSessionStateException class.....	2-14
ObjectLookupFailureException class.....	2-15
ObjectNotFoundException class.....	2-17
ObjectUpdateFailureException class .....	2-19
SessionDestroyFailureException class.....	2-21
SessionException class .....	2-23
SessionLockedException class .....	2-25
SessionLookupFailureException class.....	2-27
SessionNotFoundException class.....	2-29

## 3 oracle.owas.wrb.WRB package

WRBRunnable class.....	3-2
WRBNotRunningException class.....	3-5
WRBRuntimeException class.....	3-6

## 4 oracle.OAS.Services.ICX package

ICXRequest class .....	4-3
ICXResponse class.....	4-10
ICXInitFailedException class.....	4-15
IncompatibleWithProtocolException class.....	4-16

## 5 oracle.lang package

NLS class .....	5-2
NLSRuntimeException class .....	5-6
CharacterSetException class.....	5-7

## 6 oracle.owas.nls package

CharacterSet class.....	6-3
CharacterSetManager class.....	6-13
NLS_LANG class .....	6-15
CharacterSetException class.....	6-25
NLSRuntimeException class .....	6-26

## 7 oracle.plsql package

PBoolean class.....	7-4
PByteArray class .....	7-8
PDate class .....	7-14
PDouble class.....	7-19
PInteger class .....	7-23
PStringBuffer class .....	7-27
PValue class .....	7-38
NonUniformSizeException class .....	7-43
NullValueException class .....	7-44
PLSQLRuntimeException class.....	7-45

## 8 oracle.rdbms package

Session class .....	8-3
ServerException class .....	8-15

**A Class Hierarchy**

**B Locations of JServlet Class Files**

**Index**



---

---

# Preface

## Audience

This book is for developers who want to create Java-based Web applications for Oracle Application Server.

## The Oracle Application Server Documentation Set

This table lists the Oracle Application Server documentation set.

<b>Title of Book</b>	<b>Part No.</b>
Oracle Application Server 4.0.8 Documentation Set	A66971-03
Oracle Application Server Overview and Glossary	A60115-03
Oracle Application Server Installation Guide for Sun SPARC Solaris 2.x	A58755-03
Oracle Application Server Installation Guide for Windows NT	A58756-03
Oracle Application Server Administration Guide	A60172-03
Oracle Application Server Security Guide	A60116-03
Oracle Application Server Performance and Tuning Guide	A60120-03
Oracle Application Server Developer's Guide: PL/SQL and ODBC Applications	A66958-02
Oracle Application Server Developer's Guide: JServlet Applications	A73043-01
Oracle Application Server Developer's Guide: LiveHTML and Perl Applications	A66960-02
Oracle Application Server Developer's Guide: EJB, ECO/Java and CORBA Applications	A69966-01
Oracle Application Server Developer's Guide: C++ CORBA Applications	A70039-01
Oracle Application Server PL/SQL Web Toolkit Reference	A60123-03
Oracle Application Server PL/SQL Web Toolkit Quick Reference	A60119-03

---

<b>Title of Book</b>	<b>Part No.</b>
Oracle Application Server JServlet Toolkit Reference	A73045-01
Oracle Application Server JServlet Toolkit Quick Reference	A73044-01
Oracle Application Server Cartridge Management Framework	A58703-03
Oracle Application Server 4.0.8.1 Release Notes	A66106-04

## Conventions

This table lists the typographical conventions used in this manual.

<b>Convention</b>	<b>Example</b>	<b>Explanation</b>
bold	<b>oas.h</b> <b>owsctl</b> <b>wrbcfg</b> <b>www.oracle.com</b>	Identifies file names, utilities, processes, and URLs
italics	<i>file1</i>	Identifies a variable in text; replace this place holder with a specific value or string.
angle brackets	<filename>	Identifies a variable in code; replace this place holder with a specific value or string.
courier	owsctl start wrb	Text to be entered exactly as it appears. Also used for functions.
square brackets	[-c string] [on off]	Identifies an optional item. Identifies a choice of optional items, each separated by a vertical bar ( ), any one option can be specified.
braces	{yes no}	Identifies a choice of mandatory items, each separated by a vertical bar ( ).
ellipses	n,...	Indicates that the preceding item can be repeated any number of times.

The term “Oracle Server” refers to the database server product from Oracle Corporation.

The term “**oracle**” refers to an executable or account by that name.

The term “*oracle*” refers to the owner of the Oracle software.

---

## Technical Support Information

Oracle Global Support can be reached at the following numbers:

- In the USA: **Telephone: 1.650.506.1500**
- In Europe: **Telephone: +44 1344 860160**
- In Asia-Pacific: **Telephone: +61. 3 9246 0400**

Please prepare the following information before you call, using this page as a checklist:

- your CSI number (if applicable) or full contact details, including any special project information
- the complete release numbers of the Oracle Application Server and associated products
- the operating system name and version number
- details of error codes and numbers and descriptions. Please write these down as they occur. They are critical in helping WWCS to quickly resolve your problem.
- a full description of the issue, including:
  - **What** - What happened? For example, the command used and its result.
  - **When** -When did it happen? For example, during peak system load, or after a certain command, or after an operating system upgrade.
  - **Where** -Where did it happen? For example, on a particular system or within a certain procedure or table.
  - **Extent** - What is the extent of the problem? For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.
- Keep copies of any trace files, core dumps, and redo log files recorded at or near the time of the incident. WWCS may need these to further investigate your problem. For a list of trace and log files, see “Configuration and Log Files” in the *Administration Guide*.

For installation-related problems, please have the following additional information available:

- listings of the contents of \$ORACLE\_HOME (Unix) or %ORACLE\_HOME% (NT) and any staging area, if used.

- 
- ❑ installation logs (**install.log**, **sql.log**, **make.log**, and **os.log**) typically stored in the **\$ORACLE\_HOME/orainst** (Unix) or **%ORACLE\_HOME%\orainst** (NT) directory.

## Documentation Sales and Client Relations

In the United States:

- To order hardcopy documentation, call Documentation Sales: **1.800.252.0303**.
- For shipping inquiries, product exchanges, or returns, call Client Relations: **1.650.506.1500**.

In the United Kingdom:

- To order hardcopy documentation, call Oracle Direct Response: **+44 990 332200**.
- For shipping inquiries and upgrade requests, call Customer Relations: **+44 990 622300**.

---

## Reader's Comment Form

### Oracle Application Server 4.0 JServlet Toolkit Reference

#### Part No. A73045-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Oracle Application Server Documentation Manager  
Oracle Corporation  
500 Oracle Parkway  
Redwood Shores, CA 94065

If you would like a reply, please provide your name, address, and telephone number below:

Thank you for helping us improve our documentation.



---

---

## oracle.html package

The following tables show how HTML elements are mapped to Java classes in the oracle.html package.

### Head Elements

*Table 1–1 Mapping of head elements to Java classes*

HTML	Java class
<HEAD>	<a href="#">HtmlHead class</a>
<TITLE>	<a href="#">HtmlHead class</a> or <a href="#">HtmlPage class</a>
<STYLE>	<a href="#">Style class</a>
<SCRIPT>	<a href="#">Script class</a>
<ISINDEX>	
<BASE>	
<META>	<a href="#">MetaInfo class</a>
<LINK>	<a href="#">HeadLink class</a>

### Body Elements

*Table 1–2 Mapping of body elements to Java classes*

HTML	Java class
<BODY>	<a href="#">HtmlBody class</a>
<H1> to <H6>	<a href="#">Heading class</a>
<ADDRESS>	<a href="#">Address class</a>
<P>	<a href="#">Paragraph class</a> or <a href="#">SimpleItem class</a>

---

**Table 1–2 Mapping of body elements to Java classes**

<b>HTML</b>	<b>Java class</b>
<A href>	<a href="#">Link class</a>
<A name>	<a href="#">Anchor class</a>
<IMG>	<a href="#">Image class</a>
<IMG> (server-side image map)	<a href="#">Image class</a>
<IMG> (client-side image map)	<a href="#">ImageMap class</a>
<MAP>	<a href="#">ImageMapArea class</a>
<APPLET>	<a href="#">Applet class</a>
<PARAM>	<a href="#">Applet.addParam()</a>
 	<a href="#">LineBreak class</a>
<PRE>	<a href="#">Preformat class</a>
<BLOCKQUOTE>	<a href="#">BlockQuote class</a>
<HR>	<a href="#">HorizontalRule class</a>
<DIV>	
<CENTER>	<a href="#">Item.setAttr(ATTR_ALGN_CENTER)</a>

## List Elements

**Table 1–3 Mapping of list elements to Java classes**

<b>HTML</b>	<b>Java class</b>
<UL>	<a href="#">UnOrderedList class</a>
<OL>	<a href="#">OrderedList class</a>
<DL>	<a href="#">DefinitionList class</a>
<DIR>	<a href="#">DirectoryList class</a>
<MENU>	<a href="#">MenuList class</a>
<LI>	<a href="#">ListItem class</a>
<DT>	<a href="#">DefinitionList.addDef()</a>
<DD>	<a href="#">DefinitionList.addDef()</a>

---

## Table Elements

**Table 1–4 Mapping of table elements to Java classes**

HTML	Java class
<TABLE>	<a href="#">DynamicTable class</a>
<CAPTION>	DynamicTable.setCaption() (extended from <a href="#">Table class</a> )
<TR>	<a href="#">TableRow class</a>
<TH>	<a href="#">TableHeaderCell class</a>
<TD>	<a href="#">TableDataCell class</a>

## Text-Level Elements

**Table 1–5 Mapping of text-level elements to Java classes**

HTML	Java class
<BASEFONT>	<a href="#">BaseFont class</a>
<FONT>	<a href="#">Font class</a>
<EM>	Item.setAttr(ATTR_PHRASE_EMPHASIS) or Item.setEmphasis()
<STRONG>	Item.setAttr(ATTR_PHRASE_STRONG) or Item.setStrongEmphasis()
<DFN>	Item.setAttr(ATTR_PHRASE_DEFINITION) or Item.setDefinition()
<CODE>	Item.setAttr(ATTR_PHRASE_CODE) or Item.setCode()
<SAMP>	Item.setAttr(ATTR_PHRASE_SAMPLE) or Item.setSample()
<KBD>	Item.setAttr(ATTR_PHRASE_KEYBOARD) or Item.setKeyboard()
<VAR>	Item.setAttr(ATTR_PHRASE_VARIABLE) or Item.setVariable()
<CITE>	Item.setAttr(ATTR_PHRASE_CITATION) or Item.setCite()
<TT>	Item.setAttr(ATTR_FONT_TELETYPE) or Item.setTeletype()
<I>	Item.setAttr(ATTR_FONT_ITALIC) or Item.setItal()
<B>	Item.setAttr(ATTR_FONT_BOLD) or Item.setBold()
<U>	Item.setAttr(ATTR_FONT_UNDERLINE) or Item.setUnderline()
<STRIKE>	Item.setAttr(ATTR_FONT_STRIKE) or Item.setStrike()
<BIG>	Item.setAttr(ATTR_FONT_BIG) or Item.setFontBig()

---

**Table 1–5 Mapping of text-level elements to Java classes**

HTML	Java class
<SMALL>	Item.setAttr(ATTR_FONT_SMALL) or Item.setFontSmall()
<SUB>	Item.setAttr(ATTR_FONT_SUB) or Item.setFontSubscript()
<SUP>	Item.setAttr(ATTR_FONT_SUPER) or Item.setFontSuperscript()

## Form Elements

**Table 1–6 Mapping of form elements to Java classes**

HTML	Java class
<FORM>	Form class
<SELECT>	Select class
<OPTION>	Option class
<TEXTAREA>	TextArea class
<INPUT type=checkbox>	CheckBox class
<INPUT type=password>	PasswordField class
<INPUT type=text>	TextField class
<INPUT type=radio>	Radio class
<INPUT type=submit>	Submit class
<INPUT type=reset>	Reset class
<INPUT type=hidden>	Hidden class
<INPUT type=image>	Image class

## Other Elements

**Table 1–7 Mapping of other elements to Java classes**

HTML	Java class
<!-- ... -->	Comment class
<FRAME>	Frame class
<FRAMESET>	Frameset class
<EMBED>	Embed class
<OBJECT>	XObject class

---

This chapter contains reference pages for the following exception and error classes:

- [HtmlException class](#)
- [HtmlRuntimeException class](#)
- [HtmlInternalError class](#)

## IAlign interface

public interface **IAlign**  
Horizontal Alignment Attributes

### Constants

Name	Syntax	Description
<b>CENTER</b>	public static final int CENTER	Align to the center horizontally.
<b>LEFT</b>	public static final int LEFT	Align to the left hand side.
<b>MAX</b>	public static final int MAX	
<b>MIN</b>	public static final int MIN	
<b>NONE</b>	public static final int NONE	No horizontal alignment specified.
<b>RIGHT</b>	public static final int RIGHT	Align to the right hand side.
<b>Str</b>	public static final String Str[]	A String Array for the Attribute Values.

---

## IHtmlItem interface

public interface **IHtmlItem**

IHtmlItem is an interface that specifies the necessary functions that a class must implement to qualify as an IHtmlItem.

### See Also

- [HtmlStream class](#)
- [Item class](#)

## print

### Interface

oracle.html.IHtmlItem

### Syntax

```
public abstract void print()
```

```
public abstract void print(OutputStream out)
```

### Purpose

Prints the content of the object as an HTML string to the specified OutputStream object. If no parameter is passed then the output is to `HtmlStream.theStream()`.

### Parameters

out - the OutputStream object to be written to

## toHTML

### Interface

oracle.html.IHtmlItem

### Syntax

```
public abstract String toHTML()
```

### Purpose

Returns the content of an object as a HTML string.

---

---

**Implementation Note:** Object should be both browser-independent and browser-aware.

---

---

### Returns

An HTML string representing the content of this object.

---

## ITableFrame interface

public interface **ITableFrame**

ITableFrame encapsulates the FRAME attribute for HTML3 Tables.

### Constants

<b>Name</b>	<b>Syntax</b>	<b>Description</b>
<b>ABOVE</b>	<code>public static final int ABOVE</code>	Render the Top side of the frame.
<b>BELOW</b>	<code>public static final int BELOW</code>	Render the Bottom side of the frame.
<b>BORDER</b>	<code>public static final int BORDER</code>	Render all four sides of the frame.
<b>BOX</b>	<code>public static final int BOX</code>	Render all four sides of the frame.
<b>HSIDES</b>	<code>public static final int HSIDES</code>	Render the top and bottom sides of the frame.
<b>LHS</b>	<code>public static final int LHS</code>	Render the left hand side of the frame.
<b>RHS</b>	<code>public static final int RHS</code>	Render the right hand side of the frame.
<b>Str</b>	<code>public static final String Str[]</code>	String Array for Attribute Value.
<b>VOLD</b>	<code>public static final int VOLD</code>	Do not render any side of the frame.
<b>VSIDES</b>	<code>public static final int VSIDES</code>	Render the Left and Right hand sides of the frame.

---

## ITableRules interface

public interface **ITableRules**

ITableRules encapsulates the RULES attribute of HTML3 Table objects.

### Constants

Name	Syntax	Description
<b>ALL</b>	<code>public static final int ALL</code>	Place rules between all rows and all columns. User agents may choose to use a heavier rule between groups of rows and columns for emphasis.
<b>BASIC</b>	<code>public static final int BASIC</code>	The THEAD, TFOOT, TBODY elements divide the table into group of rows. This choice places a horizontal rule between each such group.
<b>COLS</b>	<code>public static final int COLS</code>	Place vertical rules between groups of columns as defined by COLSET and COL elements, plus horizontal rules between row groups.
<b>NONE</b>	<code>public static final int NONE</code>	Suppresses internal rulings.
<b>ROWS</b>	<code>public static final int ROWS</code>	Place horizontal rules between all rows. User agents may choose to use a heavier rule between groups of rows.
<b>Str</b>	<code>public static final String Str[]</code>	String Array for Attribute Value.

---

## ITarget interface

public interface **ITarget**

---

---

**Note:** This is a Netscape-specific interface.

---

---

ITarget defines the magic names for the TARGET attribute.

### Constants

Name	Syntax	Description
<b>BLANK</b>	public static final int BLANK	
<b>PARENT</b>	public static final int PARENT	
<b>SELF</b>	public static final int SELF	
<b>Str</b>	public static final String Str[]	
<b>TOP</b>	public static final int TOP	

### See Also

- [Frame class](#)
- [Frameset class](#)
- [Link class](#)
- [HtmlHead class](#)

## IVAlign interface

public interface **IVAlign**

Vertical alignment attributes.

### Constants

Name	Syntax	Description
<b>BASELINE</b>	<code>public static final int BASELINE</code>	(Apply to TableCell only) All cells in the same row with valign=baseline should be vertically positioned so that the first text line in each cell occur on a common baseline. The constraint does not apply to subsequent text lines in these cells.  See Also: <a href="#">TableCell class</a>
<b>BOTTOM</b>	<code>public static final int BOTTOM</code>	Align to the bottom.
<b>MAX</b>	<code>public static final int MAX</code>	
<b>MIDDLE</b>	<code>public static final int MIDDLE</code>	Align in the middle vertically.
<b>MIN</b>	<code>public static final int MIN</code>	
<b>NONE</b>	<code>public static final int NONE</code>	No Vertical Alignment Attribute specified.
<b>Str</b>	<code>public static final String Str[]</code>	A String Array for the Attribute Values.
<b>TOP</b>	<code>public static final int TOP</code>	Align to the top.

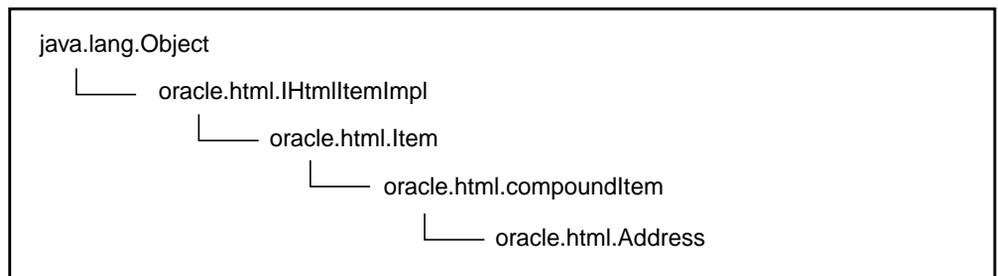
---

## Address class

public class **Address**  
extends `CompoundItem`

Address as specified in HTML. Typically, This element is rendered in italic typeface and (maybe) indented.

**Figure 1–1** *Inheritance for the Address class*



### Instructions

Use this class to add personal information such as address, signature and authorship to your page.

### Example

```
// Creates an Address object
Address addr = new Address();
// Add items to Address object
addr.addItem("My Name");
addr.addItem("My Address");
```

### In HTML

```
<ADDRESS>My NameMy Address</ADDRESS>
```

### See Also

- [CompoundItem class](#)

## Address constructor

### Class

oracle.html.Address

### Syntax

```
public Address()
```

### Description

Creates an empty Address object.

---

## Anchor class

public class **Anchor**  
extends Item

Anchor Class encapsulates an Anchor object.

---

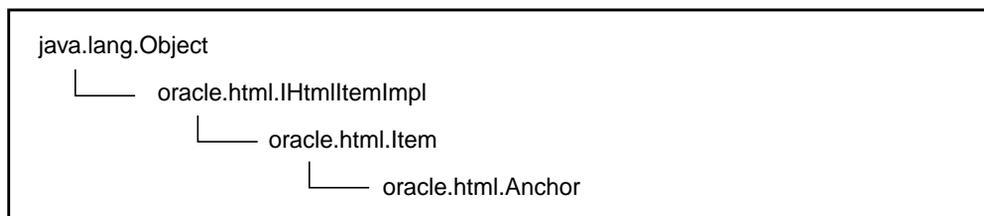
---

**Note:** This constitutes the Anchor part of ANCHOR element as specified in HTML2.

---

---

**Figure 1–2 Inheritance for the Anchor class**



### Instructions

Use this class whenever you want to create a link target. Use Link objects to create hyperlinks that point to this target.

### Example

```
// Creates an anchor
Anchor anchor = new Anchor("expire_date", new SimpleItem("Expire Date: 02/96"));
```

### In HTML

```
<A NAME="expire_date">Expire Date: 02/96</A>
```

### See Also

- [Item class](#)
- [Link class](#)

## Anchor constructor

### Class

oracle.html.Anchor

### Syntax

```
public Anchor(  
    String      name,  
    IHtmlItem  item)
```

```
public Anchor(  
    String      name,  
    String      text)
```

### Purpose

Creates a Anchor object.

### Parameters

name - the name of this anchor (becomes a link target)

item - the item to be hyperanchored

text - the text to be hyperanchored

## addItem

### Class

oracle.html.Anchor

### Syntax

```
public Anchor addItem(IHtmlItem item)
```

### Purpose

Add an item to be hyperAnchored in this object.

**See Also**

- [IHtmlItem interface](#)

**name****Class**

oracle.html.Anchor

**Syntax**

```
public String name()
```

**Returns**

The name of this Anchor.

**setName****Class**

oracle.html.Anchor

**Syntax**

```
public Anchor setName(String name)
```

**Purpose**

Sets the name of this anchor.

## toHTML

### Class

oracle.html.Anchor

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML String.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.Anchor

### Syntax

```
public String toString()
```

### Returns

The content of this object as regular string.

### Overrides

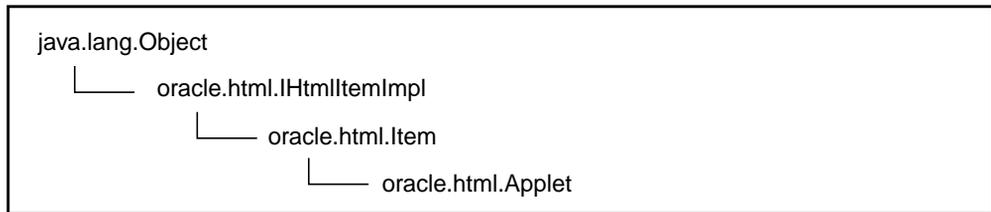
[toString](#) in [Item class](#).

## Applet class

```
public class Applet
  extends Item
  implements IValign, IAlign
```

Applet encapsulates an APPLET element as specified in JavaSoft's JDK (Java Development Kit) version 1.0.

**Figure 1–3 Inheritance for the Applet class**



### Instructions

Use this class to incorporate Java applets into your HTML page.

### Example

```
// Create a new Applet object with the following attributes:
// Applet file name: "NervousText.class"
// Width of Applet Window: 400
// Height of Applet Window: 75
// Parameter: Name="text", Value="This is an applet test"
Applet applet = new Applet("NervousText.class", 400, 75);
applet.addParam("text", "This is an applet test.");
```

### In HTML

```
<APPLET CODE="NervousText.class" WIDTH=400 HEIGHT=75>
<PARAM NAME="text" VALUE="This is an applet test.">
</APPLET>
```

**See Also**

- [XObject class](#)
- [Embed class](#)

**Applet constructor****Class**

oracle.html.Applet

**Syntax**

```
public Applet(  
    String      code,  
    int         width,  
    int         height)
```

**Purpose**

Creates an Applet object.

**Parameters**

code - applet file (e.g. "NervousText.class")

width - width of applet window in pixels

height - height of applet window in pixels

**addParam****Class**

oracle.html.Applet

### Syntax

```
public Applet addParam(  
    String      name,  
    String      value)
```

### Purpose

Adds a parameter (applet specific attribute).

### Parameters

name - name of the attribute/parameter

value - value of the parameter

## setAlign

### Class

oracle.html.Applet

### Syntax

```
public Applet setAlign(String align)
```

### Purpose

Sets the alignment attribute of the applet.

## setAltHtml

### Class

oracle.html.Applet

**Syntax**

```
public Applet setAltHtml(Item item)
```

```
public Applet setAltHtml(Container cItems)
```

**Purpose**

Sets alternative HTML Items.

**setAltText****Class**

oracle.html.Applet

**Syntax**

```
public Applet setAltText(String text)
```

**Purpose**

Sets the text that should be displayed if the browser understands the APPLET tag but cannot run Java applets.

**setCode****Class**

oracle.html.Applet

**Syntax**

```
public Applet setCode(String code)
```

### **Purpose**

Sets the name of the file that contains the applet's compiled Applet subclass.

## **setCodebase**

### **Class**

oracle.html.Applet

### **Syntax**

```
public Applet setCodebase(String codebase)
```

### **Purpose**

Sets the code base (the base URL of the applet).

## **setHeight**

### **Class**

oracle.html.Applet

### **Syntax**

```
public Applet setHeight(int height)
```

### **Purpose**

Sets the number of pixels of height of the applet window.

## setHSpace

### Class

oracle.html.Applet

### Syntax

```
public Applet setHSpace(int hspace)
```

### Purpose

Sets the number of pixels on each side of the applet window.

## setName

### Class

oracle.html.Applet

### Syntax

```
public Applet setName(String name)
```

### Purpose

Sets the instance name of the applet.

## setVSpace

### Class

oracle.html.Applet

### **Syntax**

```
public Applet setVSpace(int vspace)
```

### **Purpose**

Sets the number of pixels above and below the applet window.

## **setWidth**

### **Class**

oracle.html.Applet

### **Syntax**

```
public Applet setWidth(int width)
```

### **Purpose**

Sets the number of pixels of the width of the applet window.

## **toHTML**

### **Class**

oracle.html.Applet

### **Syntax**

```
public String toHTML()
```

### **Returns**

The content of this object as HTML string.

**Overrides**

[toHTML](#) in [Item class](#).

**toString****Class**

oracle.html.Applet

**Syntax**

```
public synchronized String toString()
```

**Returns**

A string representation of this object.

**Overrides**

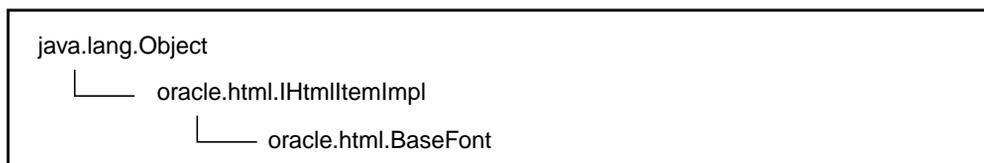
[toString](#) in [Item class](#).

## BaseFont class

```
public class BaseFont  
extends IHtmlItemImpl
```

The BaseFont class encapsulates a BASEFONT element as defined in HTML3.2.

*Figure 1-4 Inheritance for the BaseFont class*



### Instructions

Use this class to set the default value for any text in an HTML document.

### Example

```
// Creates a BaseFont object with size = 3  
BaseFont bf1 = new BaseFont(3);  
// Creates a BaseFont object with color=red and size=3  
BaseFont bf2 = new BaseFont(Color.red, null, 3);
```

### In HTML

```
<BASEFONT SIZE=3>  
<BASEFONT SIZE=3 COLOR="#FF0000">
```

## BaseFont constructor

### Class

oracle.html.BaseFont

## Syntax

```
public BaseFont(int size)
```

```
public BaseFont(  
    Color        color,  
    String       name,  
    int          size)
```

## Description

Creates a BaseFont object.

## Parameters

color - color of the base font

name - name of the base font

size - size of the base font

## toHTML

### Class

oracle.html.BaseFont

### Syntax

```
public String toHTML()
```

### Returns

Returns content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.BaseFont

### Syntax

```
public String toString()
```

### Overrides

toString in Object class.

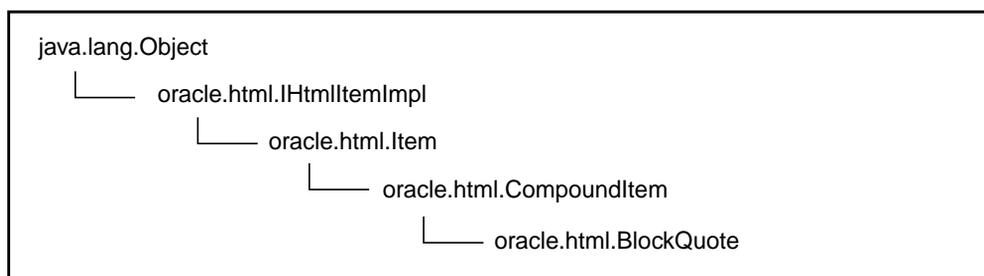
---

## BlockQuote class

public class **BlockQuote**  
extends CompoundItem

BlockQuote as specified in HTML. This usually contains text quoted from another source.

*Figure 1–5 Inheritance for the BlockQuote class*



### Instructions

Use this class to indicate that text quoted from another source is embedded in your page.

### Example

```
// Create an empty BlockQuote object
BlockQuote bq = new BlockQuote();
// Add Items to block quote object
bq.addItem("Your favorite poem");
```

### In HTML

```
<BLOCKQUOTE>Your favorite poem</BLOCKQUOTE>
```

### See Also

- [CompoundItem class](#)

## BlockQuote constructor

### Class

oracle.html.BlockQuote

### Syntax

```
public BlockQuote()
```

### Purpose

Creates an empty BlockQuote object.

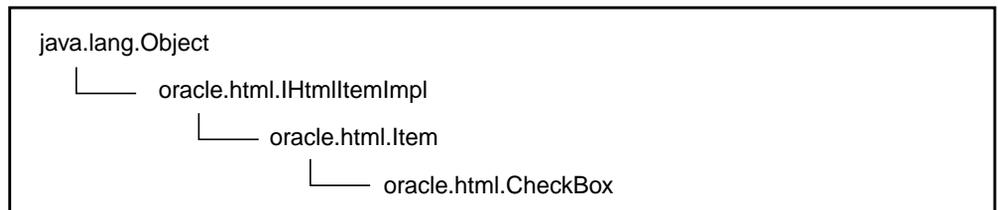
---

## CheckBox class

```
public class CheckBox  
extends Item
```

CheckBox encapsulates a check box element as specified in HTML2.

**Figure 1–6 Inheritance for the CheckBox class**



### Instructions

Use a checkbox object to present the user with a clickable box for input purposes.

### Example

```
// Create a checkbox (name="browser", value="orapb")  
CheckBox cb = new CheckBox("browser", "orapb");  
// Add checkbox to form  
form.addItem(cb);
```

### See Also

- [Item class](#)
- [Form class](#)

## CheckBox constructor

### Class

`oracle.html.CheckBox`

### **Syntax**

```
public CheckBox(  
String name,  
String value)
```

```
public CheckBox(  
String name,  
String value,  
boolean state)
```

### **Purpose**

Creates a new CheckBox object.

## **startTag**

### **Class**

oracle.html.CheckBox

### **Syntax**

```
public String startTag()
```

### **Returns**

The start tag of this object.

## **toHTML**

### **Class**

oracle.html.CheckBox

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this object as a HTML string.

**Overrides**

[toHTML](#) in [Item class](#).

## Color class

**public class Color**  
extends Object

The class Color is a helper class that provides the following features:

1. A list of predefined color values.
2. Multiple constructors that convert various Color representations to HTML color representations.

---



---

**Note:** Different browsers render colors differently, and the predefined colors may not appear as expected for all browsers.

---



---

*Figure 1–7 Inheritance for the Color class*



### Constants

Name	Syntax	Description
<b>aqua</b>	<code>public static final Color aqua</code>	The color aqua.
<b>black</b>	<code>public static final Color black</code>	The color black.
<b>blue</b>	<code>public static final Color blue</code>	The color blue.
<b>cyan</b>	<code>public static final Color cyan</code>	The color cyan.
<b>darkGray</b>	<code>public static final Color darkGray</code>	The color dark gray.
<b>fuchsia</b>	<code>public static final Color fuchsia</code>	The color fuchsia.
<b>gray</b>	<code>public static final Color gray</code>	The color gray.
<b>green</b>	<code>public static final Color green</code>	The color green.
<b>lightGray</b>	<code>public static final Color lightGray</code>	The color light gray.
<b>lime</b>	<code>public static final Color lime</code>	The color lime.

---

Name	Syntax	Description
<b>magenta</b>	<code>public static final Color magenta</code>	The color magenta.
<b>maroon</b>	<code>public static final Color maroon</code>	The color maroon.
<b>navy</b>	<code>public static final Color navy</code>	The color navy.
<b>olive</b>	<code>public static final Color olive</code>	The color olive.
<b>orange</b>	<code>public static final Color orange</code>	The color orange.
<b>pink</b>	<code>public static final Color pink</code>	The color pink.
<b>red</b>	<code>public static final Color red</code>	The color red.
<b>silver</b>	<code>public static final Color silver</code>	The color silver.
<b>teal</b>	<code>public static final Color teal</code>	The color teal.
<b>white</b>	<code>public static final Color white</code>	The color white.
<b>yellow</b>	<code>public static final Color yellow</code>	The color yellow.

---

### Example

Since `Color` does not implement `IHtmlItem`, you cannot add a color object directly to any `HtmlItem`. For certain methods that accept `Color` as argument, however, you can use either the predefined color or create your own color object and provide it as argument.

```
// Create the HTML body object
HtmlBody bd = new HtmlBody();
// Sets black as background color
bd.setBackgroundColor(Color.black);
// Sets followed link color as RGB (0,10,10)
bd.setFollowedLinkColor(new Color(0,10,10));
```

### See Also

- [HtmlPage class](#)
- [HtmlBody class](#)

## Color constructor

### Class

oracle.html.Color

### Syntax

```
public Color(String rgb)
```

```
public Color(  
    String      r,  
    String      g,  
    String      b)
```

```
public Color(  
    int         r,  
    int         g,  
    int         b)
```

```
public Color(Color color)
```

### Purpose

Creates a Color object.

### Parameters

rgb - color in format "rrggbb"

r - red color component

g - green color component

b - blue color component

color - color object from package java.awt.Color

## toString

### Class

oracle.html.Color

### Syntax

```
public String toString()
```

### Returns

Returns a string representation of this color object in the form "#rrggbb".

### Overrides

toString in Object class.

## Comment class

---

```
public class Comment  
extends IHtmlItemImpl
```

Comment as specified in HTML.

**Figure 1–8 Inheritance for the Comment class**



### Instructions

Use this class to put comments in your HTML page. Please note that comments are not visible to the client.

### Example

```
// Creates a comment  
Comment comment = new Comment("Expires: 28 Feb 1996");  
// Adds comment to body (assuming it already exists)  
body.addItem(comment);
```

### In HTML

```
<!-- Expires: 28 Feb 1996-->
```

### See Also

- [IHtmlItem interface](#)

## Comment constructor

### Class

oracle.html.Comment

### Syntax

```
public Comment(String content)
```

### Purpose

Creates a Comment object.

### Parameters

content - the comment content

## toHTML

### Class

oracle.html.Comment

### Syntax

```
public String toHTML()
```

### Returns

Returns content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Comment

### Syntax

```
public String toString()
```

### Overrides

toString in Object class.

## CompoundItem class

public class **CompoundItem**  
 extends Item

CompoundItem is a type-safe container class for Items. It allows the client to construct custom parameterized classes for component based dynamic page construction.

---



---

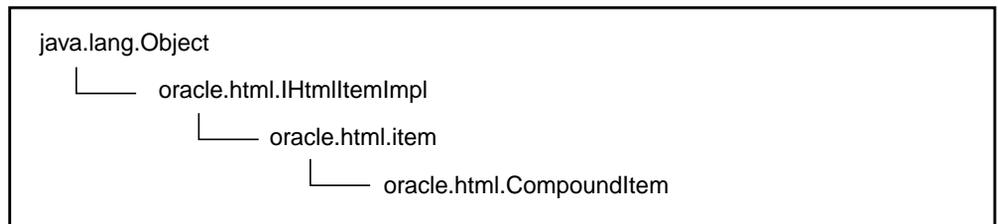
**Note:** This class differs from Container class in that a Container is not an item.

---



---

**Figure 1–9** Inheritance for the CompoundItem class



### Constants

Name	Syntax	Description
<b>INITIAL_CAPACITY</b>	<code>public static final int INITIAL_CAPACITY</code>	

### Instructions

Use this class whenever you want to create custom container components that are themselves HTML items. (e.g. methods like `setBold()` applies to this object)

### Example

```

// Create a new CompoundItem object
CompoundItem compoundItem = new CompoundItem();

// Set the default text attribute of all items in Compound Item
// Note that this operation cannot be done with a Container
compoundItem.setItal();

```

```
// Add a SimpleItem and a TextArea to the CompoundItem
compoundItem.addItem(new SimpleItem("How are you?").setBold())
                    .addItem(new TextArea("response", 30, 10));

// Add the CompoundItem to the body object (assuming it's been created)
body.addItem(compoundItem);
```

### See Also

- [Item class](#)
- [Container class](#)

## CompoundItem constructor

### Class

oracle.html.CompoundItem

### Syntax

```
public CompoundItem()
```

```
public CompoundItem(int initCapacity)
```

### Purpose

Creates a CompoundItem (with the specified storage capacity if given) object.

### Parameters

initCapacity - the initial storage capacity of this object

### See Also

- [Vector](#)

## addItem

### Class

oracle.html.CompoundItem

### Syntax

```
public synchronized CompoundItem addItem(IHtmlItem item)
```

```
public synchronized CompoundItem addItem(String string)
```

### Purpose

Adds either an item or a SimpleItem to CompoundItem, respectively.

### Parameters

item - the item to be added to this object

string - the string item to be added to this object

### Returns

this

### See Also

- [IHtmlItem interface](#)
- [SimpleItem class](#)

## itemAt

### Class

oracle.html.CompoundItem

### Syntax

```
public synchronized IHtmlItem itemAt(int index)
```

### **Purpose**

Retrieves an item from `CompoundItem`.

### **Parameters**

index - the specified position (zero-based)

### **Returns**

The item at the specified position.

## **items**

### **Class**

`oracle.html.CompoundItem`

### **Syntax**

```
public synchronized Enumeration items()
```

## **setItemAt**

### **Class**

`oracle.html.CompoundItem`

### **Syntax**

```
public synchronized CompoundItem setItemAt(  
    IHtmlItem    item,  
    int          index)
```

### **Purpose**

Sets an item at position given by index. The previous item at that position is discarded.

**Parameters**

item - what the item is to be set to.

index - the specified position

**Returns**

this

**See Also**

- Vector

**size****Class**

oracle.html.CompoundItem

**Syntax**

```
public int size()
```

**Returns**

The number of items in this object.

**toHTML****Class**

oracle.html.CompoundItem

**Syntax**

```
public synchronized String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.CompoundItem

### Syntax

```
public synchronized String toString()
```

### Returns

A String representation of this object.

### Overrides

[toString](#) in [Item](#) class.

---

## Container class

public class **Container**  
extends `IHtmlItemImpl`

Container is a type-safe container class for Items. It allows the client to construct custom parameterized classes for component based dynamic page construction.

---

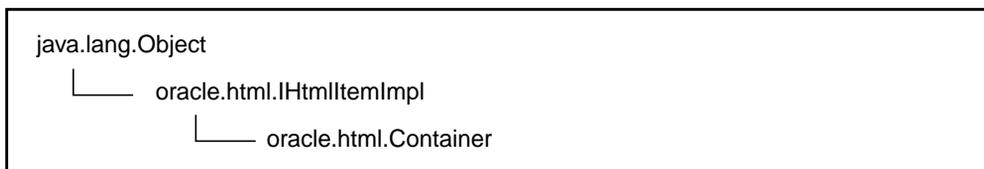
---

**Note:** A Container is not an Item.

---

---

**Figure 1–10** *Inheritance for the Container class*



### Instructions

Use this class whenever you want to create custom container components.

### Example

```
// Create a new Container object
Container container = new Container();
// Add a SimpleItem and a TextArea to the container
container.addItem(new SimpleItem("How are you?").setBold())
    .addItem(new TextArea("response", 30, 10));
// Add the container to the body object (assuming it's been created)
body.addItem(container);
```

### See Also

- [Vector](#)
- [IHtmlItem interface](#)
- [CompoundItem class](#)

## Container constructor

### Class

oracle.html.Container

### Syntax

```
public Container()
```

```
public Container(int initCapacity)
```

### Purpose

Creates a Container (with the specified storage capacity if given).

### Parameters

initCapacity - specifies the initial storage capacity of the container

### See Also

- Vector

## addItem

### Class

oracle.html.Container

### Syntax

```
public synchronized Container addItem(IHtmlItem item)
```

```
public synchronized Container addItem(String str)
```

### Purpose

Adds either an item or a SimpleItem to Container, respectively.

---

---

**Note:** Invoking this method, to add a `SimpleItem`, has the same effect as the following code fragment.

```
addItem(new SimpleItem(str));
```

---

---

### Parameters

item - the item to be added to this `Container` object

str - the `String` to be added to this `Container` object

### Returns

this

### See Also

- [IHtmlItem interface](#)
- [SimpleItem class](#)

## itemAt

### Class

oracle.html.Container

### Syntax

```
public synchronized IHtmlItem itemAt(int index)
```

### Purpose

Retrieves an item from `Container`.

### Parameters

index - the specified position (zero-based)

### Returns

The item at the specified position.

## items

### Class

oracle.html.Container

### Syntax

```
public synchronized Enumeration items()
```

## setItemAt

### Class

oracle.html.Container

### Syntax

```
public synchronized Container setItemAt(  
    IHtmlItem    item,  
    int          index)
```

### Purpose

Sets an item at position given by index. The previous item at that position is discarded.

### Parameters

item - what the item is to be set to

index - the specified position

### Returns

this

### See Also

- Vector

## size

### Class

oracle.html.Container

### Syntax

```
public int size()
```

### Returns

Returns the number of items in this Container object.

## toHTML

### Class

oracle.html.Container

### Syntax

```
public synchronized String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Container

### Syntax

```
public synchronized String toString()
```

### Returns

A String representation of this object.

### Overrides

[toString](#) in [Item](#) class.

## DefinitionList class

public class **DefinitionList**  
 extends List

DefinitionList encapsulates a DL list element in HTML.

---



---

**Note:** The HTML specifications allow multiple Definition Term DT elements to be associated with one single Definition DD element but **not** vice versa.

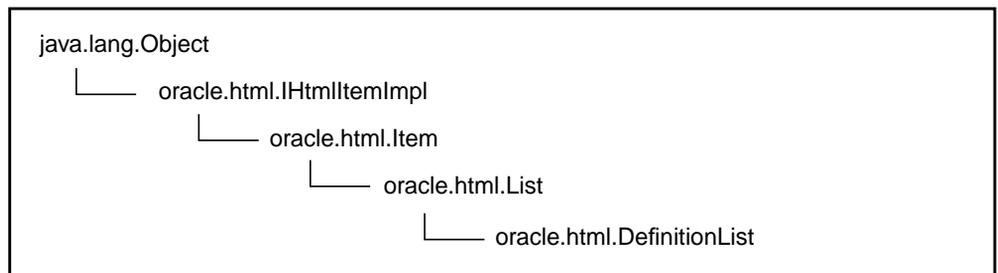
To associate multiple terms with one single definition, you need to declare a Container of definition terms, then pass the Container as the first argument to the overloaded DefinitionList constructor.

---



---

**Figure 1–11 Inheritance for the DefinitionList class**



### Example

```

// Creates a new Container object
Container dterms = new Container();
dterms.addItem(new SimpleItem("DefTerm1.1"));
dterms.addItem(new SimpleItem("DefTerm1.2"));
DefinitionList dl = new DefinitionList();

// Creates a new Definition List object, note the first argument
dl.addDef(dterms, new SimpleItem("Definition1"));

// Add definition list to body...

```

### In HTML

```
<DL>
<DT>DefTerm1.1<DT>DefTerm1.2<DD>Definition1
</DL>
```

### See Also

- [Item class](#)
- [List class](#)

## DefinitionList constructor

### Class

oracle.html.DefinitionList

### Syntax

```
public DefinitionList()

public DefinitionList(boolean compact)
```

### Purpose

Creates a DefinitionList object.

### Parameters

compact - Indicates whether compact rendering of list is desired.

## addDef

### Class

oracle.html.DefinitionList

### Syntax

```
public DefinitionList addDef(  
    IHtmlItem    defterm,  
    IHtmlItem    def
```

```
public DefinitionList addDef(  
    Container    defterms,  
    IHtmlItem    defn)
```

### Purpose

Adds a definition term to the list.

---

---

**Note:** This overloaded function expects a list/Container of definition terms and one definition. This conforms to the HTML2 specifications.

---

---

### Parameters

defterm - the term to be defined

defn - defterm's definition

### Returns

this

## toHTML

### Class

oracle.html.DefinitionList

## Syntax

```
public String toHTML()
```

## Purpose

Converts content of list object to HTML format.

## Overrides

[toHTML](#) in [List](#) class.

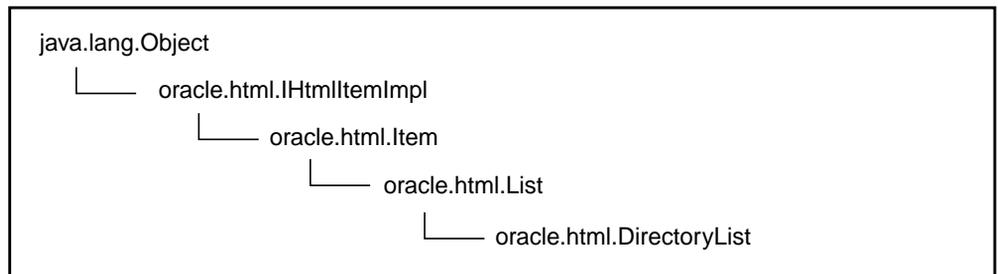
---

## DirectoryList class

```
public class DirectoryList  
extends List
```

DirectoryList encapsulates an Directory list (DIR) element in HTML.

**Figure 1–12 Inheritance for the DirectoryList class**



### Instructions

Use this class to present user with a list of items in a directory listing format.

### Example

```
// Create a Directory List  
DirectoryList drl = new DirectoryList();  
drl.addItem(new SimpleItem("Dir1"));  
drl.addItem(new SimpleItem("Dir2"));  
drl.addItem(new SimpleItem("Dir3"));
```

### In HTML

```
<DIR>  
<LI>Dir1  
<LI>Dir2  
<LI>Dir3  
</DIR>
```

### See Also

- [Item class](#)
- [List class](#)

## DirectoryList constructor

### Class

oracle.html.DirectoryList

### Syntax

```
public DirectoryList()
```

```
public DirectoryList(boolean compact)
```

### Purpose

Creates an empty DirectoryList object.

### Parameters

compact - specifies whether inter-item spacing should be reduced

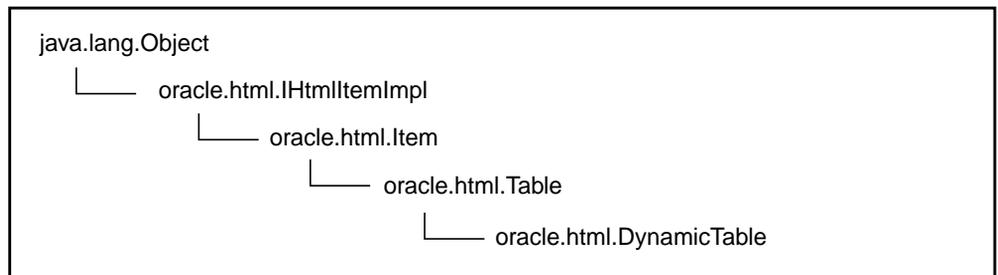
---

## DynamicTable class

```
public class DynamicTable
extends Table
```

Dynamic Table encapsulates a dynamically growable table. It mimics HTML authoring by allowing operations like `addRow()` to be applied to a DynamicTable object.

**Figure 1–13 Inheritance for the DynamicTable class**



### Instructions

Creating a table generally involves the following steps:

1. Create a DynamicTable object.
2. Create a TableRow object for each row in the table.
3. Create either a TableCell object or TableHeaderCell object for each cell in a row.
4. Add cells to each row by using the addCell method.
5. Add All rows to the table by using the addRow method.

### Example

```
// create a dynamic table with 2 columns
DynamicTable tab = new DynamicTable(2);

// create the rows and add them to the table
TableRow rows[] = new TableRow[NUM_ROWS];
for (int i=0; i< NUM_ROWS; i++) {
```

```
// allocate TableRow
rows[i] = new TableRow();
// populate row with data
rows[i].addCell(new TableHeaderCell(product.getProductID()))
        .addCell(new TableDataCell(product.getProductDescription()))
// add them to Table
tab.addRow(rows[i]);
}
```

### See Also

- [Table class](#)
- [TableRow class](#)
- [TableDataCell class](#)
- [TableHeaderCell class](#)

## DynamicTable constructor

### Class

oracle.html.DynamicTable

### Syntax

```
public DynamicTable(int cols)
```

### Purpose

Constructs a DynamicTable object.

### Parameters

cols - specifies number of columns in this table

## addRow

### Class

oracle.html.DynamicTable

### Syntax

```
public DynamicTable addRow()
```

```
public DynamicTable addRow(int numRows)
```

```
public DynamicTable addRow(TableRow row)
```

### Purpose

Adds rows to this table. If nothing is passed in, one row will be added. If a number is passed in, that number of rows will be passed in. If a TableRow is passed in, that row will be added to the table.

### Parameters

numRows - the specified number of rows

row - the new TableRow to be added to this table

### See Also

- [TableRow class](#)

## getNumRows

### Class

oracle.html.DynamicTable

### Syntax

```
public int getNumRows()
```

**Returns**

The current number of rows in this DynamicTable.

**setCellAt****Class**

oracle.html.DynamicTable

**Syntax**

```
public DynamicTable setCellAt(  
    int          col_num,  
    int          row_num,  
    SimpleItem   item)
```

**Purpose**

Sets the content of a particular cell at a specified position.

---

---

**Note:** The previous item at the position is discarded.

---

---

**Parameters**

col\_num - specifies the column number (cell count)

row\_num - specifies the row number

item - specifies the item to be set at the position

**toHTML****Class**

oracle.html.DynamicTable

**Syntax**

```
public synchronized String toHTML()
```

**Returns**

The content of this DynamicTable as a HTML string.

**Overrides**

[toHTML](#) in [Item class](#).

**toString****Class**

oracle.html.DynamicTable

**Syntax**

```
public synchronized String toString()
```

**Returns**

The content of this Dynamic Table as regular string.

**Assumptions**

1. No cell with COLSPAN or ROWSPAN attribute greater than 1.
2. All rows have the same number of cells.

**Overrides**

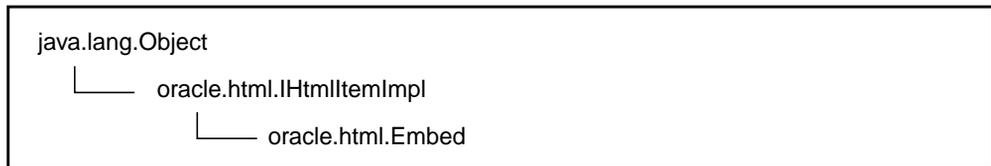
[toString](#) in [Item class](#).

## Embed class

```
public class Embed
extends IHtmlItemImpl
```

Embed encapsulates an Embed element. This allows insertion of arbitrary (e.g. multimedia) objects directly into an HTML page.

**Figure 1–14 Inheritance for the Embed class**



### Instructions

Use this class whenever you want to insert a multimedia object into your HTML page.

### Example

```
// Create a new Embed object with the following attributes:
// Embed src file name: "Myavi.avi"
// Width of Embed Window: 300
// Height of Embed Window: 200
// Parameter:  AUTOSTART = TRUE, LOOP = TRUE
Embed embed = new Embed("Myavi.avi", 300, 200);
embed.addParam("AUTOSTART", "TRUE");
embed.addParam("LOOP", "TRUE");
```

### In HTML

```
<EMBED SRC=Myavi.avi WIDTH=300 HEIGHT=200 AUTOSTART=TRUE LOOP=TRUE >
```

### See Also

- [IHtmlItem interface](#)
- [XObject class](#)

- [Applet class](#)

## Embed constructor

### Class

oracle.html.Embed

### Syntax

```
public Embed(  
    String      src,  
    int         width,  
    int         height)
```

### Purpose

Creates an Embed object.

### Parameters

src - Embed src file

width - width of Embed window in pixels

height - height of Embed window in pixels

## addParam

### Class

oracle.html.Embed

### Syntax

```
public Embed addParam(  
    String      name,  
    String      value)
```

### **Purpose**

Adds a parameter (Embed specific attribute).

### **Parameters**

name - name of the attribute/parameter

value - value of the parameter

### **Returns**

`this`

## **setHeight, setWidth**

### **Class**

`oracle.html.Embed`

### **Syntax**

```
public Embed setHeight(int height)
```

```
public Embed setWidth(int width)
```

### **Purpose**

Sets the height or width of the object window in pixels, respectively.

### **Parameters**

width - width of the object window in pixels

height - height of the object window in pixels

### **Returns**

`this`

## setSrc

### Class

oracle.html.Embed

### Syntax

```
public Embed setSrc(String src)
```

### Purpose

Set the name of the file that contains the src.

### Parameters

src - name of the file that contains the src

### Returns

this

## toHTML

### Class

oracle.html.Embed

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Embed

### Syntax

```
public synchronized String toString()
```

### Overrides

toString in Object class.

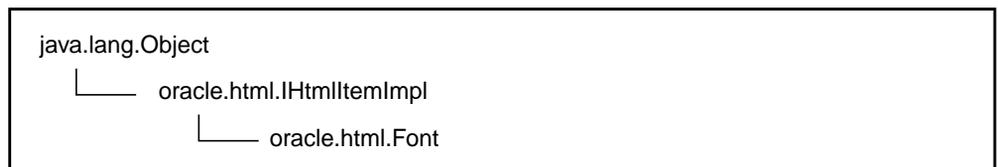
---

## Font class

```
public class Font
extends IHtmlItemImpl
```

The Font class encapsulates a FONT element as defined in HTML3.2.

**Figure 1–15 Inheritance for the Font class**



### Instructions

Use this class to set the value for any text in an HTML document.

### Example

```
// Creates a Font object with size = 3
Font f1 = new Font("3");
// Creates a Font object with color=red and size=+3
Font f2 = new Font(Color.red, null, "+3");
```

### In HTML

```
<FONT SIZE=3>
<FONT SIZE=+3 COLOR="#FF0000">
```

### See Also

- [BaseFont class](#)

## Font constructor

### Class

oracle.html.Font

### Syntax

```
public Font(int size)
```

```
public Font(String size)
```

```
public Font(  
    Color        color,  
    String       name,  
    int          size
```

```
public Font(  
    Color        color,  
    String       name,  
    String       size)
```

### Purpose

Creates a Font object.

### Parameters

color - color of the font

name - name of the font

size - size of the font

## toHTML

### Class

oracle.html.Font

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Font

### Syntax

```
public String toString()
```

### Overrides

[toString](#) in [Object](#) class.

## Form class

**public class Form**  
extends CompoundItem

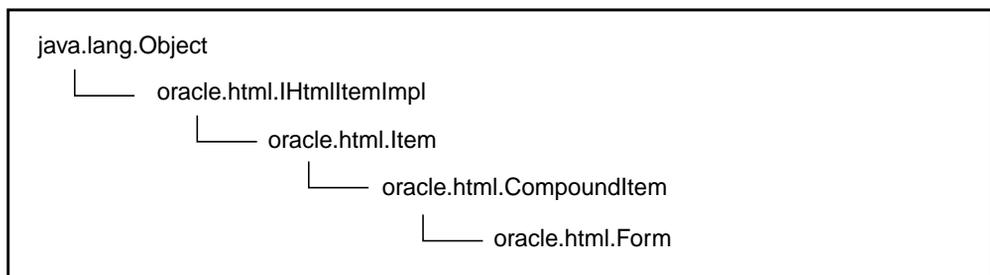
A Form is a template for a data set and an associated method and action URI. A form data set is a sequence of name/value pair fields.

Each data set is represented by one of the following objects:

- CheckBox
- Radio
- Submit
- Reset
- TextField
- TextArea
- Select
- Hidden
- PasswordField

Forms are used to allow data exchange between an user agent (the browser) and the Web Server.

**Figure 1–16 Inheritance for the Form class**



## Instructions

To create a Form object, you need to do the following:

1. Create a form object, specifying the METHOD attribute ("GET" or "POST"), and (at least) the ACTION attribute (usually an URL).
2. Populate the form object by creating at least one of the objects listed above and add it to the form using the addItem method.
3. Add the form object to the HtmlBody object.(assuming it's been created).

## Example

```
// Create a form object
Form form = new Form("GET", "http://www.myhom.com/wrb/doi");
// Create a TextField object and add it to the form
form.addItem(new TextField("textfield"));
// Add the form object to the HtmlBody object
body.addItem(form);
```

## In HTML

```
<FORM METHOD="GET" ACTION="http://www.myhom.com/wrb/doi">
<INPUT TYPE=TEXT NAME="textfield">
</FORM>
```

## See Also

- [CompoundItem class](#)
- [CheckBox class](#)
- [Radio class](#)
- [Submit class](#)
- [Reset class](#)
- [TextField class](#)
- [PasswordField class](#)
- [TextArea class](#)
- [Hidden class](#)
- [Select class](#)

- [Option class](#)
- [Frame class](#)
- [Frameset class](#)

## Form constructor

### Class

oracle.html.Form

### Syntax

```
public Form(  
    String          method,  
    String          action)
```

```
public Form(  
    String          method,  
    String          action,  
    String          target)
```

```
public Form(  
    String          method,  
    String          action,  
    String          target,  
    String          enctype)
```

### Purpose

Creates a Form object with the specified attributes.

### Parameters

method - Indicates how the form data should be sent to the server: Possible values are GET or POST. The default value is GET.

action - Specifies the URL which will be used to invoke a server-side forms handler. This can be either an HTTP server or a Mail-to URL.

**target** - Specifies to which frame should the results of the form submission be loaded into. Possible values include `_blank`, `_parent`, `_self`, `_top` and the name of the targeted window.

**enctype** - Specifies the mechanism used to encode the form's contents. Default value is "application/x-www-form-urlencoded"

## toString

### Class

oracle.html.Form

### Syntax

```
public synchronized String toString()
```

### Returns

A string representation of this object.

### Overrides

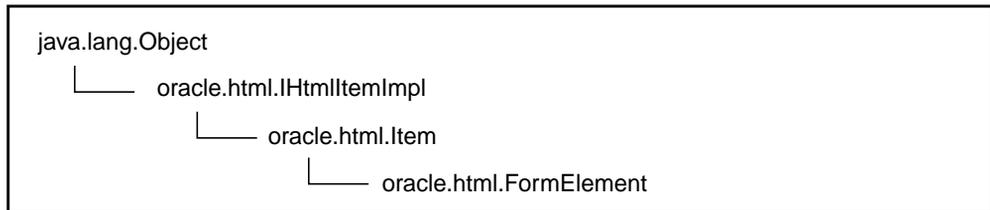
[toString](#) in [CompoundItem](#) class.

## FormElement class

**public class FormElement**  
extends Item

FormElement class encapsulates an INPUT element as specified in HTML 3.2.

**Figure 1–17 Inheritance for the FormElement class**



### Constants

Name	Syntax	Description
<b>TYPE_CHECKBOX</b>	<code>public static final int TYPE_CHECKBOX</code>	
<b>TYPE_FILE</b>	<code>public static final int TYPE_FILE</code>	
<b>TYPE_HIDDEN</b>	<code>public static final int TYPE_HIDDEN</code>	
<b>TYPE_IMAGE</b>	<code>public static final int TYPE_IMAGE</code>	
<b>TYPE_PASSWORD</b>	<code>public static final int TYPE_PASSWORD</code>	
<b>TYPE_RADIO</b>	<code>public static final int TYPE_RADIO</code>	
<b>TYPE_RESET</b>	<code>public static final int TYPE_RESET</code>	
<b>TYPE_SUBMIT</b>	<code>public static final int TYPE_SUBMIT</code>	
<b>TYPE_TEXTFIELD</b>	<code>public static final int TYPE_TEXTFIELD</code>	

### Instructions

Use this class to create FORM input fields if you don't want to use more specialized classes such as TextField and PasswordField. In particular, use this class to create FILE and IMAGE input fields.

## Example

```
// Creates a FormElement object for input=FILE
FormElement fe = new FormElement(FormElement.TYPE_FILE);
// Sets attrs name, size and accept
fe.setName("photo").setSize(20).setAccept("image/*");
```

## In HTML

```
<INPUT TYPE=FILE NAME="photo" MAXLENGTH=0 SIZE=20 ACCEPT="image/*">
```

## FormElement constructor

### Syntax

```
public FormElement(int type)
```

```
public FormElement(String type)
```

```
public FormElement(
    int          type,
    String       name,
    String       value,
    boolean      checked,
    int          size,
    int          maxlength,
    String       src,
    String       align,
    String       accept)
```

```
public FormElement(
    String       type,
    String       name,
    String       value,
    boolean      checked,
    int          size,
    int          maxlength,
    String       src,
```

```
String      align,  
String      accept)
```

### **Purpose**

Creates a FormElement object with the specified attributes.

### **Parameters**

type - specifies the type of input field { "TEXT" | "PASSWORD" | "CHECKBOX" | "RADIO" | "SUBMIT" | "RESET" | "FILE" | "HIDDEN" | "IMAGE" }

name - specifies name of this FormElement object

value - the default value of this object

checked - for radio buttons and checkboxes only

size - specifies the amount of display space allocated to this FormElement according to its type.

maxlength - specifies the number of characters that can be entered

src - for fields with background images, the URL for the background image

align - image alignment { "TOP" | "MIDDLE" | "BOTTOM" | "LEFT" | "RIGHT" }

accept - specifies the types of files that can be attached to the form's contents

## **setAccept**

### **Class**

oracle.html.FormElement

### **Syntax**

```
public FormElement setAccept(String accept)
```

### **Purpose**

Sets the ACCEPT attribute of this object.

**Parameters**

accept - specifies the types of files that can be attached to the form's contents

**Returns**

`this`

**setAlign****Class**

oracle.html.FormElement

**Syntax**

```
public FormElement setAlign(String align)
```

**Purpose**

Sets the ALIGN attribute of this object.

**Parameters**

align - image alignment { "TOP" | "MIDDLE" | "BOTTOM" | "LEFT" | "RIGHT" }

**Returns**

`this`

**setDefaultCheckValue****Class**

oracle.html.FormElement

**Syntax**

```
public FormElement setDefaultCheckValue(boolean val)
```

### **Purpose**

Sets CHECKED attribute of this object.

### **Parameters**

val - specifies whether the button should be initially checked or not. For radio buttons and checkboxes only.

### **Returns**

this

## **setDefaultValue**

### **Class**

oracle.html.FormElement

### **Syntax**

```
public FormElement setDefaultValue(String val)
```

### **Purpose**

Sets VALUE attribute of this object.

### **Parameters**

val - the default value of this object

### **Returns**

this

## **setMaxLength**

### **Class**

oracle.html.FormElement

**Syntax**

```
public FormElement setMaxLength(int mx)
```

**Purpose**

Sets MAXLENGTH attribute of this object.

**Parameters**

`maxlength` - specifies the number of characters that can be entered. If the value of `maxLength` is greater than the value of the `size` attribute, the field should scroll appropriately. The default number of characters is unlimited.

**Returns**

`this`

**setName****Class**

`oracle.html.FormElement`

**Syntax**

```
public FormElement setName(String name)
```

**Purpose**

Sets NAME attribute of this object.

**Parameters**

`name` - specifies name of this `FormElement` object

**Returns**

`this`

## setSize

### Class

oracle.html.FormElement

### Syntax

```
public FormElement setSize(int size)
```

### Purpose

Sets SIZE attribute of this object.

### Parameters

size - specifies the amount of display space allocated to this FormElement according to its type.

### Returns

this

## setSrcURL

### Class

oracle.html.FormElement

### Syntax

```
public FormElement setSrcURL(String url)
```

### Purpose

Sets SRC attribute of this object.

### Parameters

src - for fields with background images. The URL for the background image.

**Returns**

this

**toHTML****Class**

oracle.html.FormElement

**Syntax**

```
public String toHTML()
```

**Purpose**

Converts content to HTML String and returns it.

**Overrides**

[toHTML](#) in [Item](#) class.

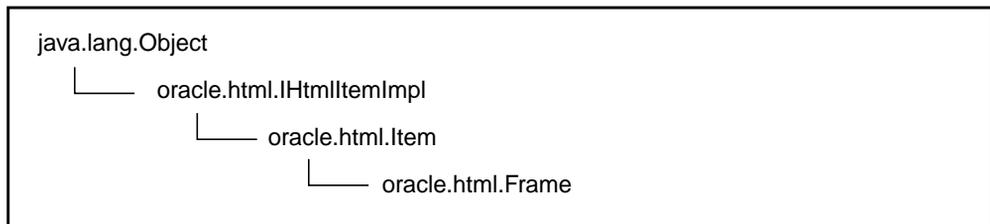
## Frame class

---

**public class Frame**  
extends Item

The Frame class encapsulates a Frame element.

**Figure 1–18 Inheritance for the Frame class**



### Instructions

Use this class together with Frame class to divide your web pages into independent, scrollable regions.

### Example

```
// Create a new Frameset object
Frameset frameset = new Frameset();
// divide the web page into 2 columns (25% width the left column, 75% the other)
frameset.setCols("25%,75%");
// set alternative Html to be displayed if browser does not support Frames
frameset.setAltHtml(altHtml);
// Creates the frame objects
Frame frames[] = new Frame[2];
frames[0]=new Frame("/java/demo.toc", "toc");
frames[1]=new Frame("/java/demo.table", "content");
// add the frames to the frameset object
for (int i=0; i < 2; i++)
    frameset.addFrame(frames[i]);
```

### In HTML

```
<FRAMESET COLS="25%,75%">
<NOFRAMES>
```

```
alt.html
</NOFRAMES>
<FRAME SRC="/java/demo.toc" NAME="toc" SCROLLING="auto">

<FRAME SRC="/java/demo.table" NAME="content" SCROLLING="auto">

</Frameset>
```

## See Also

- [Frameset class](#)

## Frame constructor

### Class

oracle.html.Frame

### Syntax

```
public Frame(String src)
```

```
public Frame(
    String      src,
    String      name)
```

```
public Frame(
    String      src,
    String      name,
    int         marginWidth,
    int         marginHeight,
    String      scrolling,
    boolean     noresize)
```

### Purpose

Creates a Frame object.

## startTag, endTag

### Class

oracle.html.Frame

### Syntax

```
public String startTag()
```

```
public String endTag()
```

### Returns

Either the start or end tag of Frame.

## setMarginHeight, setMarginWidth

### Class

oracle.html.Frame

### Syntax

```
public Frame setMarginHeight(int marginHeight)
```

```
public Frame setMarginWidth(int marginWidth)
```

## setName

### Class

oracle.html.Frame

### Syntax

```
public Frame setName(String name)
```

## setNoResize

### Class

oracle.html.Frame

### Syntax

```
public Frame setNoResize(boolean noresize)
```

## setScrolling

### Class

oracle.html.Frame

### Syntax

```
public Frame setScrolling(String scrolling)
```

## setSrc

### Class

oracle.html.Frame

### Syntax

```
public Frame setSrc(String src)
```

## toHTML

### Class

oracle.html.Frame

### Syntax

```
public String toHTML()
```

### Returns

The content of Frame as a HTML String.

### Overrides

[toHTML](#) in [Item](#) class.

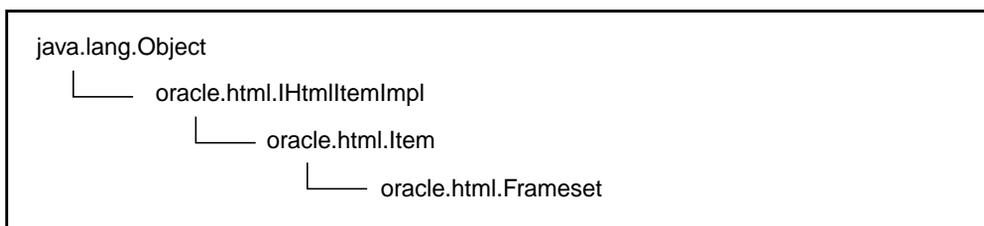
---

## Frameset class

```
public class Frameset
    extends Item
```

The Frameset class encapsulates a Frameset element.

**Figure 1–19 Inheritance for the Frameset class**



### Instructions

Use this class together with Frame class to divide your web pages into independent, scrollable regions.

### Example

```
// Create a new Frameset object
Frameset frameset = new Frameset();
// divide the web page into 2 columns (25% width the left column, 75% the other)
frameset.setCols("25%,75%");
// set alternative Html to be displayed if browser does not support Frames
frameset.setAltHtml(altHtml);
// Creates the frame objects
Frame frames[] = new Frame[2];
frames[0]=new Frame("/java/demo.toc", "toc");
frames[1]=new Frame("/java/demo.table", "content");
// add the frames to the frameset object
for (int i=0; i< 2; i++)
    frameset.addFrame(frames[i]);
```

### See Also

- [Frame class](#)

## Frameset constructor

### Class

oracle.html.Frameset

### Syntax

```
public Frameset()
```

```
public Frameset(  
    String      rows,  
    String      cols)
```

```
public Frameset(  
    String      rows,  
    String      cols,  
    CompoundItem altHtml,  
    Container   frames)
```

### Purpose

Creates a Frameset object.

---

---

**Note:** The first two constructors listed create empty frameset objects.

---

---

---

---

**Note:** You must use `setRows` or `setCols` to set either the `ROWS` or `COLS` attribute of the frameset attribute if you use the no-parameter constructor.

---

---

## addFrame

### Class

oracle.html.Frameset

### Syntax

```
public Frameset addFrame(Frame frame)
```

### Purpose

Adds a frame to frameset.

## startTag, endTag

### Class

oracle.html.Frameset

### Syntax

```
public String startTag()
```

```
public String endTag()
```

### Returns

Either the start tag or the end tag of frameset.

## setAltHtml

### Class

oracle.html.Frameset

### Syntax

```
public Frameset setAltHtml(SimpleItem altHtml)
```

```
public Frameset setAltHtml(CompoundItem altHtml)
```

### Purpose

Sets alternative HTML to be displayed which is viewable by non-frame-capable clients.

### See Also

- [SimpleItem class](#)
- [CompoundItem class](#)

## setCols, setRows

### Class

oracle.html.Frameset

### Syntax

```
public Frameset setCols(String cols)
```

```
public Frameset setRows(String rows)
```

### Purpose

Sets the COLS or ROWS attribute of this frameset, respectively.

## toHTML

### Class

oracle.html.Frameset

### Syntax

```
public String toHTML()
```

### Returns

The content of Frameset as HTML String.

### Overrides

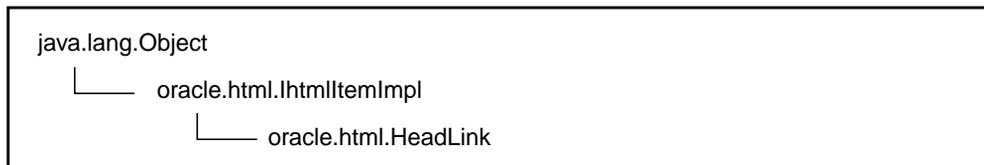
[toHTML](#) in [Item](#) class.

## HeadLink class

```
public class HeadLink  
extends IHtmlItemImpl
```

HeadLink class encapsulates a LINK element as specified in HTML 3.2.

**Figure 1–20 Inheritance for the HeadLink class**



### Instructions

A LINK element can be used to define relationships with other documents and resources (e.g. StyleSheet).

### Example

```
// Create an HeadLink object (link to a Style Sheet)
HeadLink hl = new HeadLink("http://www.xxx.yyy/ss.css", "STYLESHEET");
// Add the objects to an HtmlHead object
HtmlHead hd = new HtmlHead("Link Example");
hd.addItem(hl)
```

## HeadLink constructor

### Class

oracle.html.HeadLink

### Syntax

```
public HeadLink(  
    String href,
```

```
String      rel)

public HeadLink(
    String    href,
    String    rel,
    String    rev,
    String    title,
    String    type)
```

### **Purpose**

Creates a HeadLink object.

### **Parameters**

href - Specifies an URL designating the linked resource

rel - The forward relationship (a.k.a. "link type")

rev - Specifies a reverse relationship (a.k.a. "reverse link type")

title - Specifies an advisory title string

type - Specifies the internet media type and associated parameters for the linked document. (e.g. "text/css" for Style Sheet)

## **toHTML**

### **Class**

oracle.html.HeadLink

### **Syntax**

```
public synchronized String toHTML()
```

### **Returns**

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.HeadLink

### Syntax

```
public synchronized String toString()
```

### Overrides

toString in [Object](#) class.

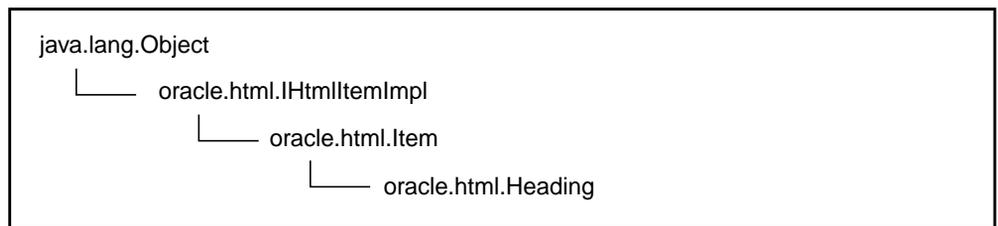
---

## Heading class

```
public class Heading  
extends Item
```

The Heading class encapsulates a Heading (H) element as defined in HTML3.2.

**Figure 1–21 Inheritance for the Heading class**



### Instructions

Use this class if you need to create an Heading element with the ALIGN attribute. Otherwise, use [setHeading\(\)](#) method in [Item class](#) instead.

### Example

```
// Creates a new H1 Heading Banner  
Heading hd = new Heading(1, "CENTER", "Hello World!");
```

### In HTML

```
<H1 ALIGN="CENTER">Hello World!</H1>
```

### See Also

- [Item class](#)

## Heading constructor

### Class

oracle.html.Heading

### Syntax

```
public Heading(  
    int          heading,  
    String       str)
```

```
public Heading(  
    int          heading,  
    IHtmlItem   item)
```

```
public Heading(  
    int          heading,  
    String       align,  
    String       str)
```

```
public Heading(  
    int          heading,  
    String       align,  
    IHtmlItem   item)
```

### Purpose

Creates a Heading object.

### Parameters

heading - the desired heading/H[1-6] of this item

align - the alignment of the Heading

item - the item to be encompassed by this heading

str - the content string of this Heading

## toHTML

### Class

oracle.html.Heading

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item class](#).

## toString

### Class

oracle.html.Heading

### Syntax

```
public String toString()
```

### Returns

A string representation of this object.

### Overrides

[toString](#) in [Item](#) class.

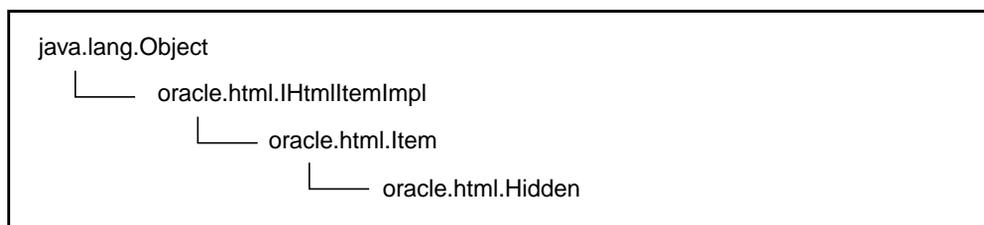
---

## Hidden class

```
public class Hidden  
extends Item
```

The Hidden class encapsulates a Hidden field. The user does not interact with this field; instead, the VALUE attribute specifies the value of the field.

**Figure 1-22** *Inheritance for the Hidden class*



### Instructions

Use this class to add a NAME/VALUE pair to be encoded in the FORM Submission URL which you don't want to be displayed.

### Example

```
// Create an Hidden object  
Hidden hd = new Hidden("hidden information id", "encrypted information");
```

### In HTML

```
<INPUT TYPE=HIDDEN NAME="hidden information id" VALUE="encrypted information">
```

### See Also

- [Form class](#)

## Hidden constructor

### Class

oracle.html.Hidden

### Syntax

```
public Hidden(  
    String      nm,  
    String      val)
```

### Purpose

Creates a Hidden object.

### Parameters

nm - name of this object

val - value of this object

## toHTML

### Class

oracle.html.Hidden

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

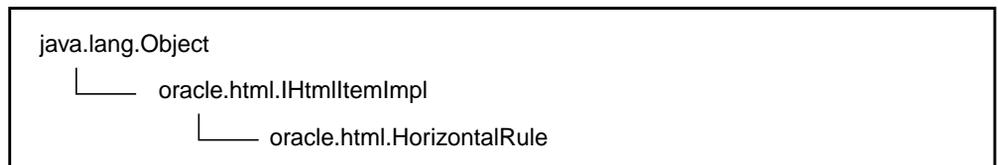
---

## HorizontalRule class

public class **HorizontalRule**  
extends IHtmlItemImpl

The HorizontalRule class encapsulates an Horizontal Rule (HR) element as defined in HTML3.2.

**Figure 1–23 Inheritance for the HorizontalRule class**



### Instructions

Use this class when you want to insert a horizontal rule in an HTML document. You should not use this class, however, if you do not need to use any of the attributes (i.e. `align`, `noshade`, `width` and `height`). In such cases, use `SimpleItem.HorizontalRule` instead.

---

---

**Hint:** A horizontal rule may be used to indicate a change in topic.

---

---

### Example

```
// Creates a HorizontalRule object with ALIGN=LEFT
HorizontalRule lb = new HorizontalRule("LEFT");
```

### See Also

- [SimpleItem class](#)

## HorizontalRule constructor

### Class

oracle.html.HorizontalRule

### Syntax

```
public HorizontalRule(  
    String          align,  
    boolean         noShade,  
    int             size,  
    String          width)
```

### Purpose

Creates a HorizontalRule object with the specified attributes.

### Parameters

`align` - the alignment (`LEFT` | `CENTER` | `RIGHT`) of the horizontal rule

`noShade` - specifies whether this rule should be rendered in a solid color

`size` - specifies the height of this rule in pixels

`width` - specifies the width of the rule in pixels or in percentage between the current left and right margins (e.g. `width = "50%"`) The default is 100%.

## toHTML

### Class

oracle.html.HorizontalRule

### Syntax

```
public String toHTML()
```

**Returns**

The content of the object as an HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl](#) class.

**toString****Class**

oracle.html.HorizontalRule

**Syntax**

```
public String toString()
```

**Overrides**

toString in Object class.

---

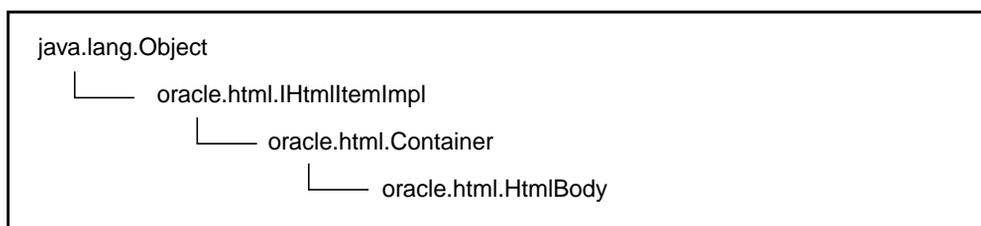
## HtmlBody class

```
public class HtmlBody
extends Container
```

HtmlBody class encapsulates the BODY part of an HTML Document.

An HtmlBody object consists of a collection/list of objects that either derive from Item class or implements IHtmlItem interface.

**Figure 1–24 Inheritance for the HtmlBody class**



### Example

It's easy to create and add items to HtmlBody, all you need to do is to create a new HtmlBody object, set the body-specific attributes, then add all the items you want to be included in the final HTML page to it:

```
// Create a new HtmlBody object
HtmlBody body = new HtmlBody();
// set attributes
body.setBackgroundImage("images/grey.gif");
// Create a new simple item
SimpleItem item = new SimpleItem("This is an item.");
// Add an item to HtmlBody object
body.addItem(item);
```

---

---

**Note:** You can easily create reusable components by creating Container objects and add them to body. Or you can simple derive your customized class from Item or Container (with parameters, for example, to customize its appearance) and add them to HtmlBody object.

---

---

### See Also

- [HtmlPage class](#)
- [HtmlHead class](#)
- [IHtmlItem interface](#)
- [Container class](#)
- [Item class](#)
- [Color class](#)

## HtmlBody constructor

### Class

oracle.html.HtmlBody

### Syntax

```
public HtmlBody()  
  
public HtmlBody(  
    String      bimg,  
    String      bcolor,  
    String      fcolor,  
    String      flinkcolor,  
    String      uflinkcolor,  
    String      alinkcolor)
```

### Purpose

Creates an empty HtmlBody object (with the specified attributes, if given).

## addItem

### Class

oracle.html.HtmlBody

### Syntax

```
public HtmlBody addItem(Container containerItem)
```

### Purpose

Adds a Container Item to this HtmlBody object.

### See Also

- [Container class](#)

## setActivatedLinkColor, setFollowedLinkColor, setUnfollowedLinkColor

### Class

oracle.html.HtmlBody

### Syntax

```
public HtmlBody setActivatedLinkColor(String color)
```

```
public HtmlBody setActivatedLinkColor(Color color)
```

```
public HtmlBody setFollowedLinkColor(String color)
```

```
public HtmlBody setFollowedLinkColor(Color color)
```

```
public HtmlBody setUnfollowedLinkColor(String color)
```

```
public HtmlBody setUnfollowedLinkColor(Color color)
```

**Purpose**

Sets one of the link color attributes (activated, followed or unfollowed) of this HtmlBody object depending on which method is called.

**setBackgroundColor, setForegroundColor****Class**

oracle.html.HtmlBody

**Syntax**

```
public HtmlBody setBackgroundColor(String color)
```

```
public HtmlBody setBackgroundColor(Color color)
```

```
public HtmlBody setForegroundColor(String color)
```

```
public HtmlBody setForegroundColor(Color color)
```

**Purpose**

Sets either the background or the foreground color attribute of this HtmlBody object.

**setBackgroundImage****Class**

oracle.html.HtmlBody

**Syntax**

```
public HtmlBody setBackgroundImage(String img)
```

### **Purpose**

Sets the background image attribute of the `HtmlBody` object.

## **toHTML**

### **Class**

`oracle.html.HtmlBody`

### **Syntax**

```
public String toHTML()
```

### **Returns**

The content of this object as a HTML string.

### **Overrides**

[toHTML](#) in [Container](#) class.

## **toString**

### **Class**

`oracle.html.HtmlBody`

### **Syntax**

```
public String toString()
```

### **Returns**

The content of this object as a regular string.

## Overrides

[toString](#) in [Container](#) class.

---

## HtmlFile class

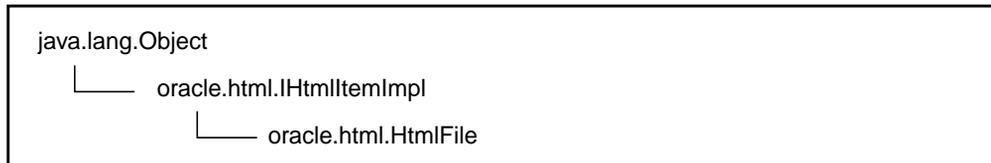
```
public class HtmlFile  
extends IHtmlItemImpl  
implements Cloneable
```

HtmlFile allows wrapping of legacy plain HTML file as IHtmlItem.

Through the use of a special tag <WRB\_INC> it also allows you to selectively replace part of the legacy HTML file with Java generated HTML (i.e. objects that implement the IHtmlItem interface)

HtmlFile is Cloneable, you may instantiate it from the content of a file, and clone it in order to apply different replacements to different clones. This is much more efficient than instantiating a new HtmlFile since the content will be preprocess only once.

**Figure 1–25 Inheritance for the HtmlFile class**



### Syntax

```
<WRB_INC NAME="name" VALUE="value">
```

### Attributes

NAME - the name/ID for this item

VALUE - the default HTML to be displayed if for some reason the item replacement mechanism has failed.

### Example

```
// Create an HtmlFile object for a file called "welcome.html"  
HtmlFile hfile = new HtmlFile("welcome.html");
```

**See Also**

- [IHtmlItem interface](#)

**HtmlFile constructor****Class**

oracle.html.HtmlFile

**Syntax**

```
public HtmlFile(String filename)
```

```
public HtmlFile(File file)
```

```
public HtmlFile(  
    File        file,  
    int         numKeys)
```

**Purpose**

Creates an HtmlFile object.

**Parameters**

file - a File object

filename - name of the file (including full path)

numKeys - number of Keys/Tags (approximately) in this file

**See Also**

- File

## clone

### Class

oracle.html.HtmlFile

### Syntax

```
public Object clone()
```

### Purpose

Clone the template page. A clone page shares the preprocess page information but have independent substitutions. Use this method when the same page needs to be rendered with different substitutions.

### Overrides

clone in Object class.

## setItemAt

### Class

oracle.html.HtmlFile

### Syntax

```
public HtmlFile setItemAt(  
    String      tag,  
    IHtmlItem  item)
```

### Purpose

Replace a named tag with the content from item.

### Parameters

tag - name of the tag to be replaced

item - an object which implements IHtmlItem interface

**See Also**

- [IHtmlItem interface](#)

**toHTML****Class**

oracle.html.HtmlFile

**Syntax**

```
public String toHTML()
```

**Returns**

The content of the object as a HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl class](#).

**toString****Class**

oracle.html.HtmlFile

**Syntax**

```
public String toString()
```

**Overrides**

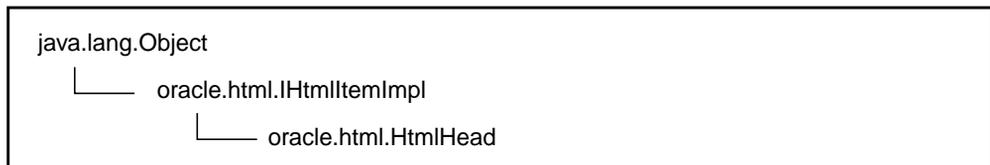
[toString](#) in [Object class](#).

## HtmlHead class

```
public class HtmlHead  
extends IHtmlItemImpl
```

Class HtmlHead encapsulates the HEAD Section of an Html Document.

**Figure 1–26 Inheritance for the HtmlHead class**



### Instructions

It's simple to create an HtmlHead object: simply invoke one of its overloaded constructors and supply the corresponding arguments. Then you can set other attributes of the object, and add it to the HtmlPage object that you've created.

### Example

```
// Create a HtmlHead object with Title = "Welcome to my home page"  
HtmlHead head = new HtmlHead("Welcome to my home page");  
// Set the Expiration date of this page  
head.addMetaInfo(new MetaInfo("Expires", "28 Feb, 1996");  
// Add this object to Htmlpage object (assuming it's been created)  
page.setHead(head);
```

### See Also

- [IHtmlItem interface](#)
- [HtmlPage class](#)
- [HtmlBody class](#)
- [HeadLink class](#)
- [URL](#)

## HtmlHead constructor

### Class

oracle.html.HtmlHead

### Syntax

```
public HtmlHead()
```

```
public HtmlHead(String title)
```

```
public HtmlHead(  
    String      title,  
    String      base,  
    String      baseTarget,  
    boolean     index)
```

### Purpose

Creates a new HtmlHead object.

### Parameters

title - the page title

base - the base HREF of this page

baseTarget - the target window for base element

index - the ISINDEX element

## addlink

### Class

oracle.html.HtmlHead

### Syntax

```
public HtmlHead addLink(IHtmlItem hi)
```

```
public HtmlHead addLink(HeadLink link)
```

### Purpose

Adds either a LINK object or a HeadLink object to this HEAD section depending on the parameter.

### Parameters

hi - An object which implements interface IHtmlItem

link - the HeadLink object to be added to this HEAD section

### Returns

this

### See Also

- [IHtmlItem interface](#)
- [HeadLink class](#)

## addMetaInfo

### Class

oracle.html.HtmlHead

### Syntax

```
public HtmlHead addMetaInfo(MetaInfo mi)
```

**Purpose**

Adds meta info. to HEAD section.

**addStyle****Class**

oracle.html.HtmlHead

**Syntax**

```
public HtmlHead addStyle(Style s)
```

**Purpose**

Adds a Style object to this HEAD section.

**Parameters**

s - the Style object to be added to the HEAD section

**Returns**

this

**See Also**

- [Style class](#)

**setBase****Class**

oracle.html.HtmlHead

### Syntax

```
public HtmlHead setBase(URL base)
```

```
public HtmlHead setBase(String base)
```

### Purpose

Sets the base URL for dereferencing relative URLs.

---

---

**Note:** If a string is passed in instead of a URL object, then it must be compliant with the W3C URL specification.

---

---

### Parameters

base - Specifies the base URL for dereferencing relative URLs

### Returns

this

### See Also

- URL

## setBaseTarget

### Class

oracle.html.HtmlHead

### Syntax

```
public HtmlHead setBaseTarget(String baseTarget)
```

### Purpose

Sets the TARGET window attribute for BASE link of this page.

## setTitle

### Class

oracle.html.HtmlHead

### Syntax

```
public HtmlHead setTitle(String title)
```

### Purpose

Sets the TITLE attribute of this page.

## toHTML

### Class

oracle.html.HtmlHead

### Syntax

```
public synchronized String toHTML()
```

### Returns

The content of this object as a HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.HtmlHead

### **Syntax**

```
public synchronized String toString()
```

### **Returns**

The content of this object as a regular string.

### **Overrides**

toString in Object class.

## HtmlPage class

```
public class HtmlPage
extends IHtmlItemImpl
implements Cloneable
```

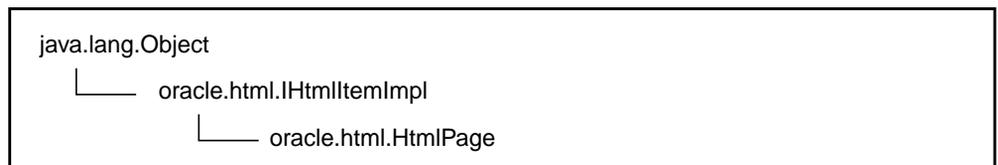
HtmlPage encapsulates an HTML Page.

There are 3 types of Html Page one can create using this package:

1. Ordinary Html Page -- consists of one HtmlHead and one HtmlBody.
2. Frame-oriented page -- consists of one HtmlHead and one Frameset.
3. Legacy Html page -- this allows you to import an existing HTML page as an HtmlPage object, plus the capability to change part of the legacy Html page.

HtmlPage is a Cloneable class. For pages loaded from files, clones can be used to have different substitutions applied to different clones but making the parsing of the input file only once, when the original instance was created.

**Figure 1–27 Inheritance for the HtmlPage class**



### Constants

Name	Syntax	Description
<b>FILE_DOC</b>	<code>public static final int FILE_DOC</code>	Legacy html page.
<b>FRAME_DOC</b>	<code>public static final int FRAME_DOC</code>	Frame based page.
<b>NORMAL_DOC</b>	<code>public static final int NORMAL_DOC</code>	Normal Html Page.

### Example

```
// Create a normal Html Page
// Create the Head object with title "Welcome to our home page"
HtmlHead head = new HtmlHead("Welcome to our home page");
```

```
// Create the Body object
HtmlBody body = new HtmlBody();
// Now creates the page
HtmlPage page = new HtmlPage(head, body);
```

### See Also

- [HtmlHead class](#)
- [HtmlBody class](#)
- [HtmlFile class](#)
- [Frameset class](#)
- [Frame class](#)
- [Color class](#)

## HtmlPage constructor

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage()
```

```
public HtmlPage(String title)
```

```
public HtmlPage(File file)
```

```
public HtmlPage(
    HtmlHead    head,
    HtmlBody    body)
```

```
public HtmlPage(
    HtmlHead    head,
    Frameset    frameset)
```

```
public HtmlPage(  
    int          type,  
    HtmlHead    head,  
    HtmlBody    body,  
    Frameset    frameset,  
    File        file)
```

## Purpose

Creates an HtmlPage object with specified parameters.

If no parameters are used the HtmlPage will be of NORMAL\_DOC with a default HtmlHead and a default empty HtmlBody.

---

---

**Note:** If FRAME\_DOC is used as the type then the body argument will be ignored.

---

---

## Parameters

type - the type of this document (NORMAL\_DOC, FRAME\_DOC, or FILE\_DOC)

head - the HEAD section of this document

body - the BODY section of this document

frameset - the FRAMESET section of this document

file - the original or legacy HTML file to be imported

## See Also

- [File](#)
- [Frame class](#)
- [Frameset class](#)
- [HtmlHead class](#)

## addItem

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage addItem(IHtmlItem item)
```

### Purpose

Adds an item to the body of this page.

### See Also

- [HtmlBody class](#)

## clone

### Class

oracle.html.HtmlPage

### Syntax

```
public Object clone()
```

### Purpose

Clones the page. Cloning defines a page with separate set of substitutions.

### Overrides

clone in Object class.

## setBody, getBody

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setBody(HtmlBody body)
```

```
public HtmlBody getBody()
```

### Purpose

Sets or retrieves the BODY element of this page.

## setHead, getHead

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setHead(HtmlHead head)
```

```
public HtmlHead getHead()
```

### Purpose

Sets or retrieves the HEAD element of this page.

## printHeader

### Class

oracle.html.HtmlPage

### Syntax

```
public void printHeader()
```

```
public void printHeader(OutputStream out)
```

### Purpose

Prints this page's header information. If an outputstream object is specified, the printing will be to that object.

### See Also

- [print](#) in [IHtmlItemImpl](#) class.

## setActivatedLinkColor, setFollowedLinkColor, setUnfollowedLinkColor

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setActivatedLinkColor(String color)
```

```
public HtmlPage setActivatedLinkColor(Color color)
```

```
public HtmlPage setFollowedLinkColor(String color)
```

```
public HtmlPage setFollowedLinkColor(Color color)
```

```
public HtmlPage setUnfollowedLinkColor(String color)
```

```
public HtmlPage setUnfollowedLinkColor(Color color)
```

### Purpose

Sets one of the link color attributes (activated, followed or unfollowed) of this document depending on which method is called.

## setBackgroundcolor, setForegroundColor

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setBackgroundcolor(String color)
```

```
public HtmlPage setBackgroundcolor(Color color)
```

```
public HtmlPage setForegroundColor(String color)
```

```
public HtmlPage setForegroundColor(Color color)
```

### Purpose

Sets either the background or the foreground color attribute of this document.

## setBackgroundImage

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setBackgroundImage(String img)
```

### Purpose

Sets the background image attribute of this document.

## setItemAt

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setItemAt(  
    String      tag,  
    IHtmlItem  item)
```

### Purpose

Replaces a specified tag in the legacy Html page with IHtmlItem item.

---

---

**Note:** This method is only valid for a document of type FRAME\_DOC.

---

---

### See Also

- [HtmlFile class](#)

## setTitle

### Class

oracle.html.HtmlPage

### Syntax

```
public HtmlPage setTitle(String title)
```

### Purpose

Sets the title of this page.

**Parameters**

title - Specifies the title of this document

**See Also**

- [HtmlHead class](#)

**toHTML****Class**

oracle.html.HtmlPage

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this page as a HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl class](#).

**toString****Class**

oracle.html.HtmlPage

**Syntax**

```
public String toString()
```

### **Returns**

The content of this page as a regular string.

### **Overrides**

toString in Object class.

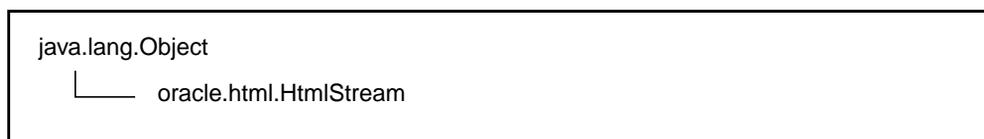
---

## HtmlStream class

```
public class HtmlStream  
extends Object
```

The `HtmlStream` class encapsulates an output stream which contains a stream of characters to be sent back to the user agent.

*Figure 1–28 Inheritance for the `HtmlStream` class*



### Instructions

In most cases, you should not use this class directly. Use the [HtmlPage class](#) and the other relevant component classes instead.

### Example

```
HtmlStream myStream = HtmlStream.theStream();  
myStream.println("Content-type: text/html");  
myStream.println();
```

## print

### Class

`oracle.html.HtmlStream`

### Syntax

```
public synchronized void print(IHtmlItem item)
```

```
public synchronized void print(Object obj)
```

```
public synchronized void print(String str)
```

```
public synchronized void print(  
    String      str,  
    boolean     bUnicode)
```

### **Purpose**

Prints a string.

### **Parameters**

item - the item to be printed

obj - the object to be printed

str - the string to be printed

bUnicode - specifies whether the lower 8 bit of each character should be discarded. If bUnicode = true, all 16 bits of each character are printed out

## **println**

### **Class**

oracle.html.HtmlStream

### **Syntax**

```
public synchronized void println()
```

```
public synchronized void println(String str)
```

```
public synchronized void println(  
    String      str,  
    boolean     bUnicode)
```

### **Purpose**

Prints a string followed by a new line. If no string is specified, only a new line is printed.

### Parameters

`str` - the string to be printed

`bUnicode` - specifies whether the lower 8 bit of each character should be discarded. If `bUnicode = true`, all 16 bits of each character are printed out.

## setUnicodeOutput, unicodeOutput

### Class

oracle.html.HtmlStream

### Syntax

```
public synchronized void setUnicodeOutput(boolean bUnicode)
```

```
public boolean unicodeOutput()
```

### Purpose

Sets or retrieves the value of the internal flag that indicates whether Unicode output is preferred over ASCII output.

---

---

**Note:** Unicode characters are not ASCII compatible.

---

---

### Parameters

`bUnicode` - if true, character Strings will be printed in Unicode format (fixed 16-bit)

### Returns

`setUnicodeOutput` returns `true` if character strings will be printed in Unicode format and `false` otherwise.

### See Also

- [print](#)

## theStream

### Class

oracle.html.HtmlStream

### Syntax

```
public static HtmlStream theStream()
```

### Returns

The singleton instance of HtmlStream.

## write

### Class

oracle.html.HtmlStream

### Syntax

```
public synchronized void write(int b) throws IOException
```

```
public synchronized void write(byte b[]) throws IOException
```

```
public synchronized void write(  
    byte        b[],  
    int         offset,  
    int         length) throws IOException
```

### Purpose

Writes either a byte (integer), an array of bytes (b[]) or a sub array of bytes.

### Parameters

b - the byte or byte array

offset - the start offset in the byte array

length - the number of bytes to be written

**Throws**

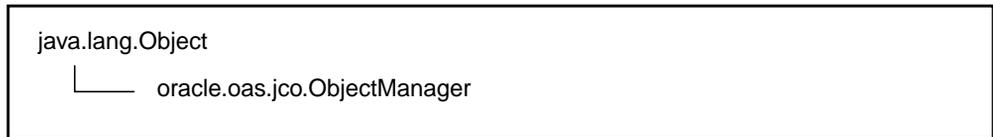
IOException if an I/O error has occurred.

## HtmlStringBuffer class

```
public class HtmlStringBuffer
    extends Object
    implements IHttpItem
```

HtmlStringBuffer class mimics class java.lang.StringBuffer. The primary addition in terms of functionality is the ability to conveniently add quoted strings to the buffer.

*Figure 1–29 Inheritance for the HtmlStringBuffer class*



### Variables

Name	Syntax	Description
<b>DOUBLE_QUOTE_CHAR</b>	public static char DOUBLE_QUOTE_CHAR	Double quote character.
<b>NEWLINE_CHAR</b>	public static char NEWLINE_CHAR	New line character.
<b>SINGLE_QUOTE_CHAR</b>	public static char SINGLE_QUOTE_CHAR	Single quote character.

### See Also

- StringBuffer

## HtmlStringBuffer constructor

### Class

oracle.html.HtmlStringBuffer

### Syntax

```
public HtmlStringBuffer()
```

```
public HtmlStringBuffer(int length)
```

```
public HtmlStringBuffer(String str)
```

### **Purpose**

Creates a new `HtmlStringBuffer` object instance. The parameters define other attributes of the buffer.

### **Parameters**

`length` - the initial capacity of the buffer

`str` - the initial string for this buffer

## **append**

### **Class**

`oracle.html.HtmlStringBuffer`

### **Syntax**

```
public synchronized HtmlStringBuffer append(boolean b)
```

```
public synchronized HtmlStringBuffer append(int i)
```

```
public synchronized HtmlStringBuffer append(char ch)
```

```
public synchronized HtmlStringBuffer append(String str)
```

```
public synchronized HtmlStringBuffer append(char str[])
```

```
public synchronized HtmlStringBuffer append(  
    char          str[],  
    int           offset,  
    int           length)
```

```
public synchronized HtmlStringBuffer append(  
    String          str,  
    char           quoteChar)  
  
public synchronized HtmlStringBuffer append(  
    String          str,  
    boolean        quote)  
  
public synchronized HtmlStringBuffer append(Object obj)  
  
public synchronized HtmlStringBuffer append(IHtmlItem item)
```

### **Purpose**

Appends the string representation of the first parameter. See the parameter list for more information.

### **Parameters**

**b** - the boolean value to be appended

**i** - the integer to be appended

**ch** - the character to be appended

**str** - the string or character array to be appended

**offset** - the index of the first character to be appended

**length** - the number of characters to be appended

**quoteChar** - the quote character to surround the string argument with in the appended string

**quote** - specifies whether the string should be quoted using double quote

**obj** - the object whose string representation are to be appended to this buffer

**item** - the object whose HTML string representation are to be appended to this buffer

### **Returns**

`this`

## appendln

### Class

oracle.html.HtmlStringBuffer

### Syntax

```
public synchronized HtmlStringBuffer appendln(String str)
```

### Purpose

Appends the string to the string buffer and add a newline character after the string characters.

### Parameters

str - the string to be appended to the string buffer

### Returns

this

## getChars

### Class

oracle.html.HtmlStringBuffer

### Syntax

```
public synchronized char[] getChars()
```

```
public synchronized void getChars(  
    int          srcBegin,  
    int          srcEnd,  
    char        dst[],  
    int          dstBegin)
```

### **Purpose**

If no parameters are used, a new character array is allocated and returned that contains a copy of the characters in this string buffer. Otherwise, the string buffer is copied into the character array `dst`.

### **Returns**

`getChars()` only: A new character array containing the contents of this string buffer.

## **length**

### **Class**

`oracle.html.HtmlStringBuffer`

### **Syntax**

```
public int length()
```

### **Returns**

The number of characters in this string buffer.

## **print**

### **Class**

`oracle.html.HtmlStringBuffer`

### **Syntax**

```
public void print()
```

```
public void print(OutputStream out)
```

## toHTML

### Class

oracle.html.HtmlStringBuffer

### Syntax

```
public String toHTML()
```

## toString

### Class

oracle.html.HtmlStringBuffer

### Syntax

```
public String toString()
```

### Purpose

A new `String` object is allocated and initialized to contain the character sequence currently represented by this string buffer.

### Returns

A string representation of the string buffer.

### Overrides

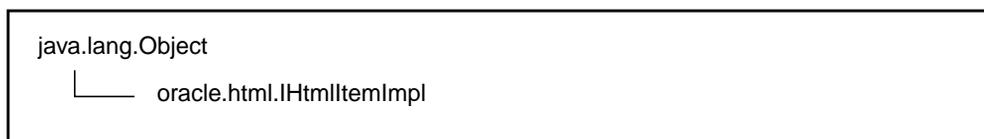
`toString` in `Object` class.

## IHtmlItemImpl class

```
public abstract class IHtmlItemImpl  
extends Object  
implements IHtmlItem
```

IHtmlItemImpl class provides standard implementation for interface IHtmlItem.

**Figure 1–30** Inheritance for the IHtmlItemImpl class



### Instructions

If you want to create your own class that implements the IHtmlItem interface yet you don't want to derive your class from one of the base classes (e.g. [Container class](#), [Item class](#)) in this package, then it may make sense to derive your class from this class.

It has the advantages of low overload and efficient implementation. Of course, you should still implement the method [toHTML\(\)](#) yourself.

### See Also

- [IHtmlItem interface](#)
- [Item class](#)

## IHtmlItemImpl constructor

### Class

```
oracle.html.IHtmlItemImpl
```

### Syntax

```
public IHtmlItemImpl().
```

## print

### Class

oracle.html.IHtmlItemImpl

### Syntax

```
public synchronized void print()
```

```
public synchronized void print(OutputStream out)
```

### Purpose

Prints the content of this object as HTML string to the specified OutputStream object. If no OutputStream object is given `HtmlStream.theStream()` is used.

---

---

**Note:** By default, this method calls method `toHTML` to get the string representation.

---

---

### Parameters

out - the OutputStream object to be written to

### See Also

- [theStream](#) in [HtmlStream](#) class

## toHTML

### Class

oracle.html.IHtmlItemImpl

### Syntax

```
public abstract synchronized String toHTML()
```

## Returns

An HTML string representing the content of this object.

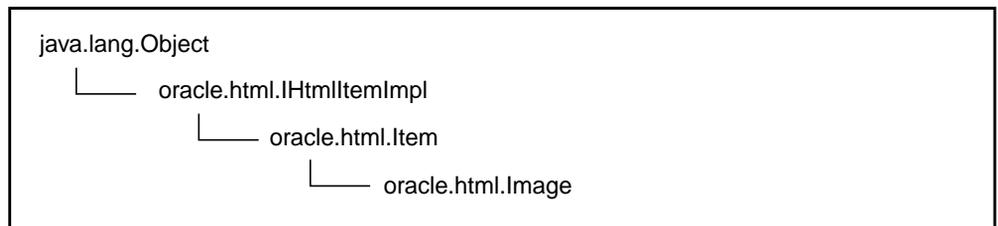
---

## Image class

public class **Image**  
extends `Item`  
implements `IValign`, `IHAlign`

Class `Image` encapsulates an `IMG` element in HTML2.

**Figure 1–31 Inheritance for the Image class**



### Instructions

Use this class whenever you want to include an image or a server-side image map in your dynamic HTML page.

### Example

```
// Create an image
Image img1 = new Image("res/home_icon.gif");
// Create an ImageMap
Image img2 = new Image("res/home_map.gif", "This is an image map",
    IValign.TOP, true);
```

### In HTML

```
<IMG SRC="res/home_icon.gif">
<IMG SRC="res/home_map.gif" ALT="This is an image map" ALIGN=TOP ISMAP>
```

### See Also

- [IHAlign interface](#)
- [IValign interface](#)

- [Item class](#)
- [ImageMap class](#)
- [ImageMapArea class](#)

## Image constructor

### Class

oracle.html.Image

### Syntax

```
public Image(String imgURI)
```

```
public Image(  
    String      imgURI,  
    String      altTxt,  
    int         align,  
    boolean     ismap)
```

```
public Image(  
    String      imgURI,  
    String      altTxt,  
    int         align,  
    boolean     ismap,  
    String      usemap)
```

```
public Image(  
    String      imgURI,  
    String      altTxt,  
    int         halign,  
    int         valign,  
    int         width,  
    int         height,  
    int         border,  
    int         hspace,  
    int         vspace,  
    boolean     ismap,
```

String usemap)

## Purpose

Creates an Image object.

## Parameters

imgURI - the URI for the image file

altTxt - alternative text to be displayed if user agent cannot display the image

align - vertical alignment of the image

halign - horizontal alignment of the image

valign - vertical alignment of the image

width - specifies the intended width of the image in pixels

height - specifies the intended height of the image in pixels

border - specifies the width of border in pixels

hspace - specifies the width of white space to the left and right of the image in pixels

vspace - specifies the height of white space above and below the image in pixels

ismap - specifies whether this is an ImageMap. Setting this attribute causes the location to be passed to the server.

usemap - specifies whether Client Side ImageMap is used

## See Also

- [IAlign interface](#)
- [VAlign interface](#)
- [Item class](#)
- [ImageMap class](#)
- [ImageMapArea class](#)

## startTag

### Class

oracle.html.Image

### Syntax

```
public String startTag()
```

### Returns

The start tag of this object.

## toHTML

### Class

oracle.html.Image

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.Image

**Syntax**

```
public String toString()
```

**Returns**

A string representation of this object.

**Overrides**

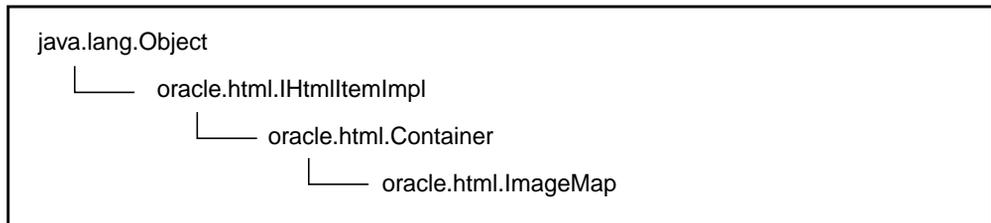
[toString](#) in [Item class](#).

## ImageMap class

**public class ImageMap**  
extends Container

The ImageMap class encapsulates a client-side image map element.

**Figure 1–32 Inheritance for the Logger class**



### Example

```
// Create a new ImageMap (named "map1")
ImageMap map = new ImageMap("map1");
// Specify a circular region which, when clicked, will lead to "index.html"
map.addArea(new ImageMapArea("CIRCLE", "107,153,80", "index.html"));
// Create a new Image object that uses ImageMap named "map1"
Image img = new Image("res/home_icon.gif", "alternative text", IValign.NONE,
false, "#map1");
```

### In HTML

```
<MAP NAME="map1">
<AREA SHAPE="CIRCLE" COORDS="107,153,80" HREF="index.html">
</MAP>
<IMG SRC="res/home_icon.gif" ALT=alternative text USEMAP="#map1"></BODY>
```

### See Also

- [ImageMapArea class](#)
- [Image class](#)

## ImageMap constructor

### Class

oracle.html.ImageMap

### Syntax

```
public ImageMap(String name)
```

### Purpose

Creates a client side image map object.

### Parameters

name - name of image map

## addArea

### Class

oracle.html.ImageMap

### Syntax

```
public ImageMap addArea(ImageMapArea area)
```

### Purpose

Adds an ImageMapArea to this object.

### Parameters

area - the area to be included

## toHTML

### Class

oracle.html.ImageMap

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [Container](#) class.

## toString

### Class

oracle.html.ImageMap

### Syntax

```
public String toString()
```

### Overrides

[toString](#) in [Container](#) class.

---

## ImageMapArea class

```
public class ImageMapArea
extends IHtmlItemImpl
```

The ImageMapArea class encapsulates an area element in a client side image map.

---

---

**Note:** This is not part of standard HTML yet.

---

---

*Figure 1–33 Inheritance for the Logger class*



### Example

```
// Create a new ImageMap (named "map1")
ImageMap map = new ImageMap("map1");
// Specify a circular region which, when clicked, will lead to "index.html"
map.addArea(new ImageMapArea("CIRCLE", "107,153,80", "index.html"));
// Create a new Image object that uses ImageMap named "map1"
Image img = new Image(img,"alternative test",IAlign.NONE, false, "#map1");
```

### See Also

- [ImageMap class](#)
- [Image class](#)

## ImageMapArea constructor

### Class

oracle.html.ImageMapArea

## Syntax

```
public ImageMapArea(  
    String      shape,  
    String      coords,  
    String      href)
```

```
public ImageMapArea(  
    String      shape,  
    String      coords,  
    String      href,  
    String      altTxt)
```

## Purpose

Creates a client side image map area object.

## Parameters

shape - the shape of this area { RECT | POLY | CIRCLE | DEFAULT } (optional)

coords - the coordinates of this area (optional)

href - where a click in the area should lead (optional)

alt - alternative text to be displayed

## See Also

- [ImageMap class](#)

## toHTML

### Class

oracle.html.ImageMapArea

### Syntax

```
public String toHTML()
```

**Returns**

The content of object as HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl](#) class.

**toString****Class**

oracle.html.ImageMapArea

**Syntax**

```
public String toString()
```

**Overrides**

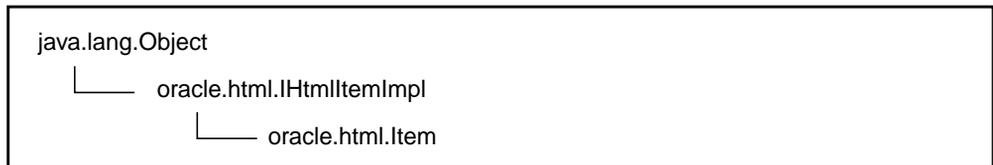
toString in [Object](#) class.

## Item class

public abstract class **Item**  
extends IHtmlItemImpl

The Item class is an abstract base class that provides the basic markup functionality for all HTML item classes.

**Figure 1–34 Inheritance for the Item class**



### Constants

Name	Syntax	Description
<b>ATTR_ALGN_CENTER</b>	public static final int ATTR_ALGN_CENTER	
<b>ATTR_BIG</b>	public static final int ATTR_BIG	
<b>ATTR_BOLD</b>	public static final int ATTR_BOLD	
<b>ATTR_CITATION</b>	public static final int ATTR_CITATION	
<b>ATTR_CODE</b>	public static final int ATTR_CODE	
<b>ATTR_DEFINITION</b>	public static final int ATTR_DEFINITION	
<b>ATTR_EMPHASIS</b>	public static final int ATTR_EMPHASIS	
<b>ATTR_FONT_BIG</b>	public static final int ATTR_FONT_BIG	
<b>ATTR_FONT_BOLD</b>	public static final int ATTR_FONT_BOLD	
<b>ATTR_FONT_ITALIC</b>	public static final int ATTR_FONT_ITALIC	
<b>ATTR_FONT_SMALL</b>	public static final int ATTR_FONT_SMALL	
<b>ATTR_FONT_STRIKE</b>	public static final int ATTR_FONT_STRIKE	
<b>ATTR_FONT_SUB</b>	public static final int ATTR_FONT_SUB	
<b>ATTR_FONT_SUPER</b>	public static final int ATTR_FONT_SUPER	

Name	Syntax	Description
<b>ATTR_FONT_TELETYPE</b>	<code>public static final int ATTR_FONT_TELETYPE</code>	
<b>ATTR_FONT_UNDERLINE</b>	<code>public static final int ATTR_FONT_UNDERLINE</code>	
<b>ATTR_ITALIC</b>	<code>public static final int ATTR_ITALIC</code>	
<b>ATTR_KEYBOARD</b>	<code>public static final int ATTR_KEYBOARD</code>	
<b>ATTR_PHRASE_CITATION</b>	<code>public static final int ATTR_PHRASE_CITATION</code>	
<b>ATTR_PHRASE_CODE</b>	<code>public static final int ATTR_PHRASE_CODE</code>	
<b>ATTR_PHRASE_DEFINITION</b>	<code>public static final int ATTR_PHRASE_DEFINITION</code>	
<b>ATTR_PHRASE_EMPHASIS</b>	<code>public static final int ATTR_PHRASE_EMPHASIS</code>	
<b>ATTR_PHRASE_KEYBOARD</b>	<code>public static final int ATTR_PHRASE_KEYBOARD</code>	
<b>ATTR_PHRASE_SAMPLE</b>	<code>public static final int ATTR_PHRASE_SAMPLE</code>	
<b>ATTR_PHRASE_STRONG</b>	<code>public static final int ATTR_PHRASE_STRONG</code>	
<b>ATTR_PHRASE_VARIABLE</b>	<code>public static final int ATTR_PHRASE_VARIABLE</code>	
<b>ATTR_SAMPLE</b>	<code>public static final int ATTR_SAMPLE</code>	
<b>ATTR_SMALL</b>	<code>public static final int ATTR_SMALL</code>	
<b>ATTR_STRIKE</b>	<code>public static final int ATTR_STRIKE</code>	
<b>ATTR_STRONG</b>	<code>public static final int ATTR_STRONG</code>	
<b>ATTR_SUB</b>	<code>public static final int ATTR_SUB</code>	
<b>ATTR_SUPER</b>	<code>public static final int ATTR_SUPER</code>	
<b>ATTR_TELETYPE</b>	<code>public static final int ATTR_TELETYPE</code>	
<b>ATTR_UNDERLINE</b>	<code>public static final int ATTR_UNDERLINE</code>	
<b>ATTR_VARIABLE</b>	<code>public static final int ATTR_VARIABLE</code>	

## Instructions

Since this is an abstract class, you cannot instantiate an object from this class directly. Instead, you should create your own class that extends Item class and then instantiate that new class.

## See Also

- [IHtmlItem interface](#)

- [SimpleItem class](#)
- [StyleElement class](#)

## Item constructor

### Class

oracle.html.Item

### Syntax

```
public Item()  
  
public Item(  
    boolean    citation,  
    boolean    code,  
    boolean    emphasis,  
    boolean    keyboard,  
    boolean    sample,  
    boolean    strongEmphasis,  
    boolean    variable,  
    boolean    italic,  
    boolean    bold,  
    boolean    teletype,  
    int        heading)
```

### Purpose

Creates an item object with specified attributes.

### Parameters

citation - specifies whether the CITE/citation attribute should be set

code - Specifies whether the CODE attribute should be set

emphasis - specifies whether the EM/Emphasis attribute should be set

keyboard - specifies whether the KBD/Keyboard attribute should be set

sample - specifies whether the SAMP/Sample attribute should be set

**strongEmphasis** - specifies whether the STRONG attribute should be set

**variable** - specifies whether the VAR/Variable attribute should be set

**italic** - specifies whether the I/Italic attribute should be set

**bold** - specifies whether the B/Bold attribute should be set

**teletype** - specifies whether the TT/Teletype attribute should be set

**heading** - specifies the value of H/Heading attribute (value: 0-6). Use value 0 if you don't want to set this attribute

## clearAttr, setAttr

### Class

oracle.html.Item

### Syntax

```
public Item clearAttr(int attr)
```

```
public Item setAttr(int attr)
```

```
public Item setAttr(  
    int          attr,  
    boolean     val)
```

### Purpose

Clears or sets the specified attribute for this object.

### Parameters

**attr** - the index for the attribute (e.g. ATTR\_BOLD)

**val** - the value of the attribute

### Returns

`this`

## setBold

### Class

oracle.html.Item

### Syntax

```
public Item setBold()
```

```
public Item setBold(boolean val)
```

### Purpose

Sets or clears the Bold/B attribute of item. If no parameter is passed the attribute is set.

## setCenter

### Class

oracle.html.Item

### Syntax

```
public Item setCenter()
```

```
public Item setCenter(boolean val)
```

### Purpose

Sets or clears the center attribute of item. If no parameter is passed the attribute is set.

## setCite

### Class

oracle.html.Item

### Syntax

```
public Item setCite()
```

```
public Item setCite(boolean val)
```

### Purpose

Sets or clears Citation/CITE attribute of item. If no parameter is passed the attribute is set.

## setCode

### Class

oracle.html.Item

### Syntax

```
public Item setCode()
```

```
public Item setCode(boolean val)
```

### Purpose

Sets or clears the Code/CODE attribute of item. If no parameter is passed the attribute is set.

## setDefinition

### Class

oracle.html.Item

### Syntax

```
public Item setDefinition()
```

```
public Item setDefinition(boolean val)
```

### Purpose

Sets or clears the Definition/DFN attribute of item. If no parameter is passed the attribute is set.

## setEmphasis

### Class

oracle.html.Item

### Syntax

```
public Item setEmphasis()
```

```
public Item setEmphasis(boolean val)
```

### Purpose

Sets or clears Emphasis/EM attribute of item. If no parameter is passed the attribute is set.

## setFontBig

### Class

oracle.html.Item

### Syntax

```
public Item setFontBig()
```

```
public Item setFontBig(boolean val)
```

### Purpose

Sets or clears font BIG attribute of item. If no parameter is passed the attribute is set.

## setFontColor

### Class

oracle.html.Item

### Syntax

```
public Item setFontColor(Color color)
```

### Purpose

Sets font color attribute of item.

### Parameters

color - the desired of this item

### Returns

this

### See Also

- [Color class](#)

## setFontSize

### Class

oracle.html.Item

### Syntax

```
public Item setFontSize(int size)
```

### Purpose

Sets font size attribute of item.

### Parameters

size - the desired size of this item

### Returns

this

## setFontSmall

### Class

oracle.html.Item

### Syntax

```
public Item setFontSmall()
```

```
public Item setFontSmall(boolean val)
```

**Purpose**

Sets or clears the font SMALL attribute of item. If no parameter is passed the attribute is set.

**setFontSubscript, setFontSuperscript****Class**

oracle.html.Item

**Syntax**

```
public Item setFontSubscript()
```

```
public Item setFontSubscript(boolean val)
```

```
public Item setFontSuperscript()
```

```
public Item setFontSuperscript(boolean val)
```

**Purpose**

Sets or clears font SUB (subscript) or SUP (superscript) attribute of item. If no parameter is passed the attribute is set.

**setHeading****Class**

oracle.html.Item

**Syntax**

```
public Item setHeading(int i)
```

### **Purpose**

Sets Heading/H[1-6] attribute of item.

### **Parameters**

i - the desired Heading/H[1-6] of this item

### **Returns**

this

## **setItal**

### **Class**

oracle.html.Item

### **Syntax**

```
public Item setItal()
```

```
public Item setItal(boolean val)
```

### **Purpose**

Sets or clears the Italic/I attribute of item. If no parameter is passed the attribute is set.

## **setKeyboard**

### **Class**

oracle.html.Item

### **Syntax**

```
public Item setKeyboard()
```

```
public Item setKeyboard(boolean val)
```

**Purpose**

Sets or clears Keyboard/KBD attribute of item. If no parameter is passed the attribute is set.

**setSample****Class**

oracle.html.Item

**Syntax**

```
public Item setSample()
```

```
public Item setSample(boolean val)
```

**Purpose**

Sets or clears the Sample/SAMP attribute of item. If no parameter is passed the attribute is set.

**setStrike****Class**

oracle.html.Item

**Syntax**

```
public Item setStrike()
```

```
public Item setStrike(boolean val)
```

### **Purpose**

Sets or clears Strike/**STRIKE** attribute of item. If no parameter is passed the attribute is set.

## **setStrongEmphasis**

### **Class**

oracle.html.Item

### **Syntax**

```
public Item setStrongEmphasis()
```

```
public Item setStrongEmphasis(boolean val)
```

### **Purpose**

Sets or clears StrongEmphasis/**STRONG** attribute of item. If no parameter is passed the attribute is set.

## **setStyleElement**

### **Class**

oracle.html.Item

### **Syntax**

```
public Item setStyleElement(StyleElement elem)
```

## Purpose

Associates a particular style with this Item object.

---

---

**Note:** You can define a style through creating and setting properties of a StyleElement object.

---

---

## Returns

this

## See Also

- [StyleElement class](#)

# setTeletype

## Class

oracle.html.Item

## Syntax

```
public Item setTeletype()
```

```
public Item setTeletype(boolean val)
```

## Purpose

Sets or clears the Teletype/TT attribute of item. If no parameter is passed the attribute is set.

## setUnderline

### Class

oracle.html.Item

### Syntax

```
public Item setUnderline()
```

```
public Item setUnderline(boolean val)
```

### Purpose

Sets or clears the Underline/U attribute of item. If no parameter is passed the attribute is set.

## setVariable

### Class

oracle.html.Item

### Syntax

```
public Item setVariable()
```

```
public Item setVariable(boolean val)
```

### Purpose

Sets or clears the Variable/VAR attribute of item. If no parameter is passed the attribute is set.

## toHTML

### Class

oracle.html.Item

### Syntax

```
public String toHTML()
```

### Returns

An HTML string representing the content of this object.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Syntax

```
public String toString()
```

### Returns

A string representation of this object.

### Overrides

toString in Object class.

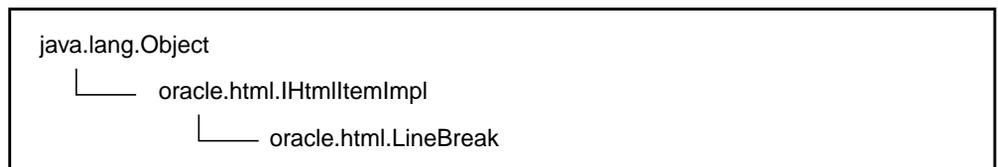
---

## LineBreak class

```
public class LineBreak  
extends IHtmlItemImpl
```

The LineBreak class encapsulates a LineBreak (BR) element as defined in HTML3.2.

*Figure 1–35 Inheritance for the LineBreak class*



### Instructions

Use this class when you want to insert a line break. If you do not use the CLEAR attribute, you should use SimpleItem.LineBreak instead.

### Example

```
// Creates a LineBreak object with CLEAR=LEFT  
LineBreak lb = new LineBreak("LEFT");
```

### See Also

- [SimpleItem class](#)

## LineBreak constructor

### Syntax

```
public LineBreak(String clear)
```

### Purpose

Creates a LineBreak object.

### Parameters

clear - the CLEAR attribute can be used to move down past floating images on either margin CLEAR=LEFT moves down past floating images on the left margin CLEAR=RIGHT does the same for floating images on the right margin CLEAR=ALL does the same for such images for both left and right margins

## toHTML

### Class

oracle.html.LineBreak

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.LineBreak

### Syntax

```
public String toString()
```

### Overrides

[toString](#) in [Object](#) class.

---

## Link class

public class **Link**  
extends Item

Link Class encapsulates a HyperLink object.

---

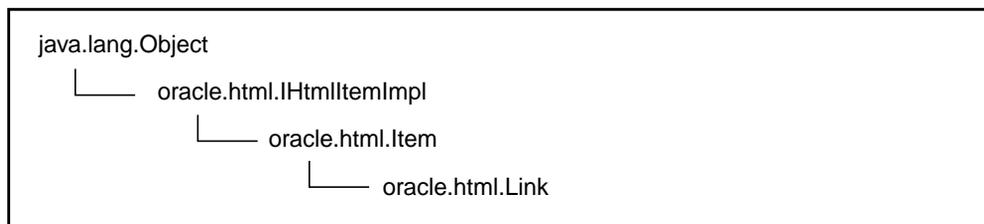
---

**Note:** This only constitutes the link part of ANCHOR element as specified in HTML2.

---

---

*Figure 1–36 Inheritance for the Link class*



### Instructions

Use this class to add hyperlinks to your HTML page.

### Example

```
// Create a simple Link pointing to URL "http://www.oracle.com"
Link link = new Link ("http://www.oracle.com", "Go to Oracle's Home Page");
```

### In HTML

```
<A HREF="http://www.oracle.com">Go to Oracle's Home Page</A>
```

### See Also

- [Item class](#)
- [Frame class](#)
- [Frameset class](#)

## Link constructor

### Class

oracle.html.Link

### Syntax

```
public Link(  
    URL          url,  
    IHtmlItem    item)
```

```
public Link(  
    Anchor       anchor,  
    IHtmlItem    item)
```

```
public Link(  
    String       url,  
    String       text)
```

```
public Link(  
    String       url,  
    IHtmlItem    item)
```

```
public Link(  
    String       url,  
    IHtmlItem    item,  
    String       target)
```

### Purpose

Creates a Link object.

### Parameters

url - the url for the hyperlink

item - the item to be hyperlink activated

text - specifies the text to be hyperlinked

anchor - an anchor object (the target of this link)

target - the target window

## addItem

### Class

oracle.html.Link

### Syntax

```
public Link addItem(IHtmlItem item)
```

### Purpose

Add an item to be hyperlinked in this object.

### Parameters

item - specifies the item to be hyperlinked

## toHTML

### Class

oracle.html.Link

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.Link

### Syntax

```
public synchronized String toString()
```

### Returns

The content of this object as regular string.

### Overrides

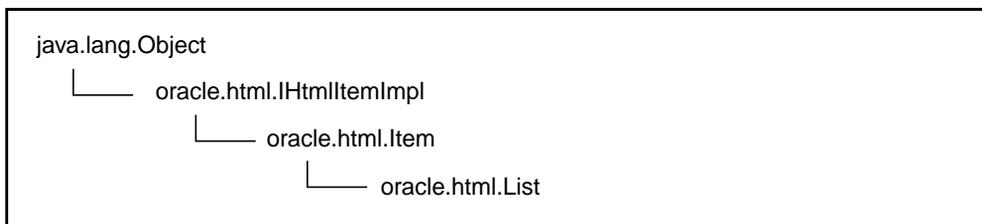
[toString](#) in [Item](#) class.

## List class

public class **List**  
extends Item

The List class is the base class for all List classes as specified in HTML 3.2. You can create an object of any type using this class or create an object from one of the derived classes.

**Figure 1–37 Inheritance for the List class**



### Constants and Variables

Name	Syntax	Description
<b>DEFINITION</b>	public static final int DEFINITION	Definition list.
<b>DIRECTORY</b>	public static final int DIRECTORY	Directory list.
<b>MENU</b>	public static final int MENU	Menu list
<b>ORDERED</b>	public static final int ORDERED	Ordered list
<b>UNORDERED</b>	public static final int UNORDERED	UnOrdered list

### Instructions

Use this class to create any list object and add list items to it.

---

**Note:** Don't use this class for creating Definition List -- use class DefinitionList instead.

---

### Examples

```
// Creates an Ordered List
```

```
List ol = new List(List.ORDERED, false);  
// Add an item to newly created list object  
ol.addItem("Miranda Padgett");  
ol.addItem("Karla Jameson");
```

### In HTML

```
<OL>  
<LI>Miranda Padgett  
<LI>Karla Jameson  
</OL>
```

### See Also

- [Item class](#)
- [CompoundItem class](#)
- [DirectoryList class](#)
- [DefinitionList class](#)
- [MenuList class](#)
- [OrderedList class](#)
- [UnOrderedList class](#)

## List constructor

### Class

oracle.html.List

### Syntax

```
public List(  
    int                type,  
    boolean            compact)
```

## Purpose

Creates an empty List object.

---

---

**Note:** This constructor is not part of this class' public interface.

---

---

## Parameters

type - specifies type ( DEFINITION | DIRECTORY | MENU | ORDERED | UNORDERED ) of list to be created.

compact - specifies whether inter-item spacing is to be reduced.

## addItem

### Class

oracle.html.List

### Syntax

```
public synchronized List addItem(IHtmlItem item)
```

```
public synchronized List addItem(String string)
```

### Purpose

Adds an item to this List object.

### Parameters

item - the item to be added to this object

### Returns

this

### See Also

- [IHtmlItem interface](#)
- [SimpleItem class](#)

## setItemAt, itemAt

### Class

oracle.html.List

### Syntax

```
public synchronized List setItemAt(  
    IHtmlItem    item,  
    int          index)
```

```
public synchronized IHtmlItem itemAt(int index)
```

### Purpose

Sets an item at position given by index or retrieves an item from the List object.

---

---

**Note:** When setting items the previous item at that position is discarded.

---

---

### Parameters

item - what the item is to be set to

index - the specified position

### Returns

this

## size

### Class

oracle.html.List

### Syntax

```
public int size()
```

**Returns**

The number of items in this list.

**toHTML****Class**

oracle.html.List

**Syntax**

```
public String toHTML()
```

**Purpose**

Converts content of list object to HTML format. Item separator is <LI> for non-List-Item type objects.

**Returns**

A formatted HTML string representing the content of this list object.

**Overrides**

[toHTML](#) in [Item class](#).

**toString****Class**

oracle.html.List

**Syntax**

```
public String toString()
```

### **Purpose**

Converts content of List object to string format. All items in the list are separated by a comma ",".

### **Returns**

The string representation of this list object.

### **Overrides**

[toString](#) in [Item class](#).

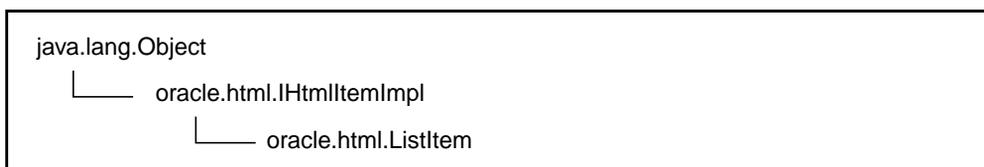
---

## ListItem class

```
public class ListItem  
extends IHtmlItemImpl
```

The ListItem class encapsulates a ListItem (LI) element as defined in HTML3.2.

**Figure 1–38 Inheritance for the ListItem class**



### Instructions

Use this class when you want to insert an item into a list. If you do not use either the TYPE or VALUE attributes, then you do not need to use this class to insert an item into a list. Use the generic addItem method from class List (i.e. CompoundItem) instead.

### Example

```
// Creates a ListItem object with TYPE=a and value=1  
ListItem li = new ListItem("a", 1, "This is a ListItem object");  
// Creates an OrderedList object  
OrderedList ol = new OrderedList();  
// Add the list item to the OrderedList object  
ol.addItem(li);
```

### In HTML

```
<OL>  
<LI TYPE=a VALUE=1>This is a ListItem object  
</OL>
```

### See Also

- [IHtmlItem interface](#)

- [CompoundItem class](#)
- [List class](#)
- [DirectoryList class](#)
- [DefinitionList class](#)
- [MenuList class](#)
- [OrderedList class](#)
- [UnOrderedList class](#)

## ListItem constructor

### Syntax

```
public ListItem(String str)
```

```
public ListItem(IHtmlItem item)
```

```
public ListItem(  
    String      type,  
    int        value,  
    String      item)
```

```
public ListItem(  
    String      type,  
    int        value,  
    IHtmlItem  item)
```

### Purpose

Creates a ListItem object.

### Parameters

type - the TYPE attribute can be used to change the bullet style in unordered lists and the numbering style in ordered list

value - use this attribute to reset sequence number

item - an item to be associated with this list item

## toHTML

### Class

oracle.html.ListItem

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class

## toString

### Class

oracle.html.ListItem

### Syntax

```
public String toString()
```

### Overrides

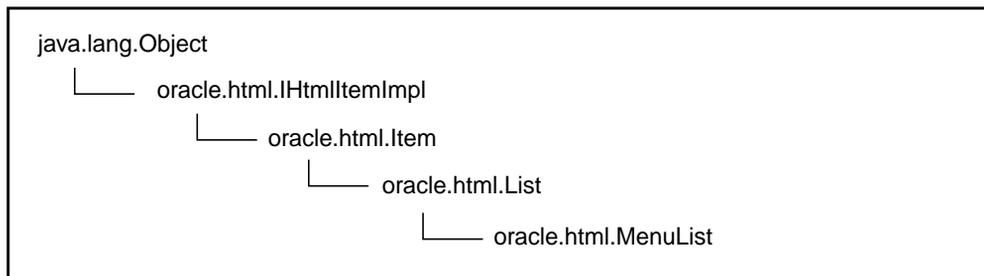
[toString](#) in [Object](#) class.

## MenuList class

```
public class MenuList  
extends List
```

MenuList encapsulates an MENU list element in HTML.

**Figure 1–39** Inheritance for the MenuList class



### Instructions

Use this class to present a list of items in a menu-like style to the user. The menu style is usually more compact than that of the unordered list.

### Example

```
// Create a MenuList object  
MenuList menulist = new MenuList();  
// Add new items to the list  
menulist.addItem(new SimpleItem("Menu Item 1"))  
    .addItem(new SimpleItem("Menu Item 2"));  
// Add the list object to the body (assuming it already exists)  
body.addItem(menulist);
```

### In HTML

```
<MENU>  
<LI>Menu Item 1  
<LI>Menu Item 2  
</MENU>
```

**See Also**

- [Item class](#)
- [List class](#)

**MenuList constructor****Class**

oracle.html.MenuList

**Syntax**

```
public MenuList()
```

```
public MenuList(boolean compact)
```

**Purpose**

Creates an empty MenuList object.

**Parameters**

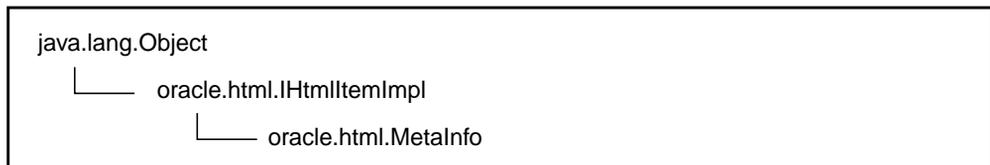
compact - specifies whether inter-item spacing should be reduced

## MetaInfo class

```
public class MetaInfo  
extends IHtmlItemImpl
```

MetaInfo class encapsulates META element as specified in HTML2.

**Figure 1–40 Inheritance for the MetaInfo class**



### Instructions

To create a MetaInfo object, invoke its constructor and supply the necessary parameters. Then add the object to the HtmlHead object.

### Example

```
// Create a MetaInfo object  
MetaInfo meta = new MetaInfo("Expires", "28 Feb, 1996");  
// Add MetaInfo object to HtmlHead head (assuming it's been created)  
head.addMetaInfo(meta);
```

### In HTML

```
<META HTTP-EQUIV="Expires" CONTENT="28 Feb, 1996">
```

### See Also

- [Item class](#)

## MetaInfo constructor

### Class

oracle.html.MetaInfo

### Syntax

```
public MetaInfo(  
    String      equiv,  
    String      content)
```

```
public MetaInfo(  
    String      equiv,  
    String      name,  
    String      content,  
    String      url)
```

### Purpose

Creates a MetaInfo object.

### Parameters

equiv - binds the element to an HTTP header field. An HTTP server may use this information when processing this document

name - meta-information name

content - specifies the value/content of this header field

url - the document's url

## toHTML

### Class

oracle.html.MetaInfo

### Syntax

```
public String toHTML()
```

### Returns

The content of object as HTML string.

### Overrides

[toHTML](#) in [IHtmlItemImpl](#) class

## toString

### Class

oracle.html.MetaInfo

### Syntax

```
public String toString()
```

### Overrides

toString in Object class.

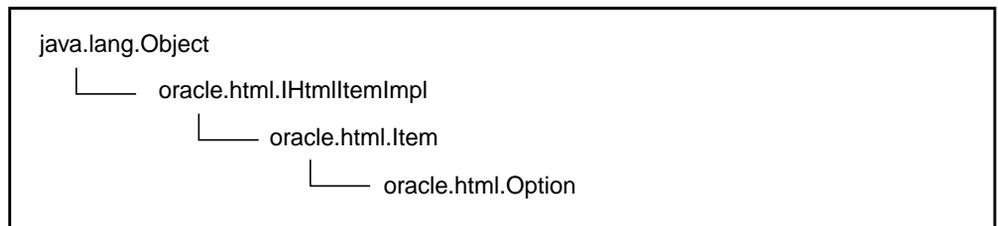
---

## Option class

public class **Option**  
 extends Item

The Option class encapsulates the OPTION element in HTML2. This element can only occur within a select statement. It represents one choice.

**Figure 1-41 Inheritance for the Option class**



### Instructions

Use this class in conjunction with the Select class to present the user an enumerated list of values.

### Example

```

// Creates a Select object with 5 choices.
Form frm = new Form("GET", "http://www.myhom.com/wrb/doit");
frm
    .addItem(new SimpleItem("Which server do you use?").setBold())
    .addItem(new Select("server")
        .addOption(new Option("Oracle WebServer"))
        .addOption(new Option("Netscape"))
        .addOption(new Option("Apache"))
        .addOption(new Option("Mosaic"))
        .addOption(new Option("CERN")));
  
```

### In HTML

```

<FORM METHOD="GET" ACTION="http://www.myhom.com/wrb/doit"><B>Which server do
you use?</B><SELECT NAME="server">
<OPTION>Oracle WebServer
<OPTION>Netscape
  
```

```
<OPTION>Apache  
<OPTION>Mosaic  
<OPTION>CERN  
  
</SELECT>  
</FORM>
```

### See Also

- [Select class](#)

## Option constructor

### Class

oracle.html.Option

### Syntax

```
public Option(String txt)
```

```
public Option(  
    String      txt,  
    String      val,  
    boolean     state)
```

### Purpose

Creates an Option object with content and specified attributes.

## toHTML

### Class

oracle.html.Option

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this object as HTML string.

**Overrides**

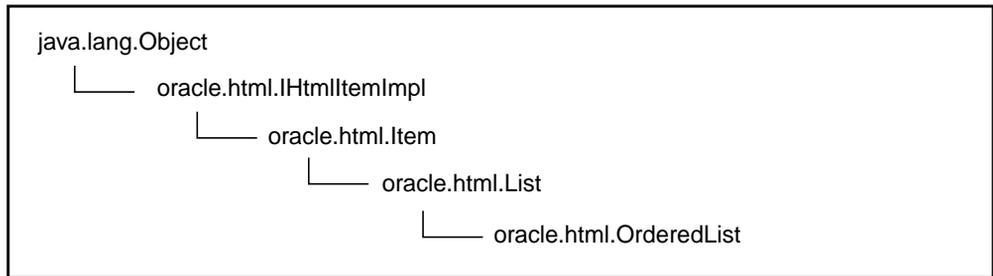
[toHTML](#) in [Item class](#)

## OrderedList class

**public class OrderedList**  
 extends List

OrderedList encapsulates an Ordered (OL) list element in HTML.

**Figure 1–42 Inheritance for the OrderedList class**



### Instructions

Use this class whenever you want to present the user with a list of items in an ordered manner.

### Example

```

// Create an Ordered List
OrderedList ol = new OrderedList();
ol.addItem(new SimpleItem("Ordered1"));
ol.addItem(new SimpleItem("Ordered2"));
ol.addItem(new SimpleItem("Ordered3"));
    
```

### See Also

- [Item class](#)
- [List class](#)

## OrderedList constructor

### Class

oracle.html.OrderedList

### Syntax

```
public OrderedList()
```

```
public OrderedList(  
    String          type,  
    int             start,  
    boolean         compact)
```

### Purpose

Creates an OrderedList object (with the specified attributes, if given).

### Parameters

type - specifies numbering style

start - starting sequence number

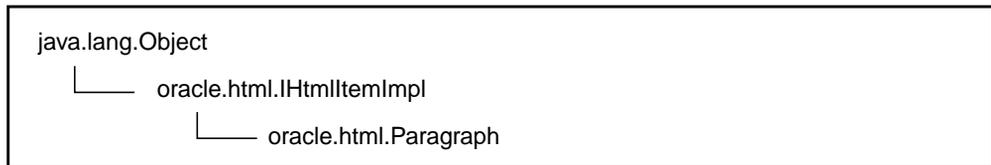
compact - specifies whether inter-item spacing should be reduced

## Paragraph class

```
public class Paragraph  
extends IHtmlItemImpl
```

The Paragraph class encapsulates a Paragraph (P) element as defined in HTML3.2.

*Figure 1–43 Inheritance for the Paragraph class*



### Instructions

Use this class when you want to insert a paragraph break. If you do not use the ALIGN attribute, you should use SimpleItem.Paragraph instead.

### Example

```
// Creates a Paragraph object with ALIGN=LEFT  
Paragraph lb = new Paragraph("LEFT");
```

### See Also

- [SimpleItem class](#)

## Paragraph constructor

### Class

oracle.html.Paragraph

### Syntax

```
public Paragraph(String align)
```

**Purpose**

Creates a Paragraph object.

**Parameters**

align - the alignment of the paragraph

**toHTML****Class**

oracle.html.Paragraph

**Syntax**

```
public String toHTML()
```

**Returns**

The content of the object as an HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Paragraph

### Syntax

```
public String toString()
```

### Overrides

toString in Object class.

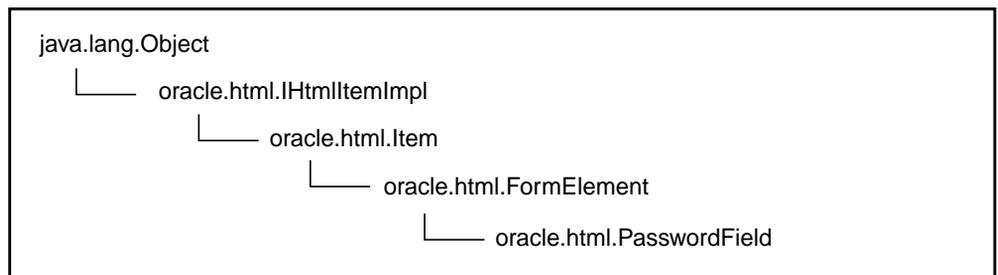
---

## PasswordField class

public class **PasswordField**  
extends FormElement

A PasswordField is the same as a text field element, except that the value is obscured as it is entered.

**Figure 1–44** Inheritance for the PasswordField class



### Instructions

Use this class to present user with a text entry field for password entry purposes.

### Example

```
// Create a PasswordField object
Form frm = new Form("GET", "http://www.myhom.com/wrb/doit");
PasswordField passwd = new PasswordField("passwd");
frm.addItem(passwd);
```

### In HTML

```
<FORM METHOD="GET" ACTION="http://www.myhom.com/wrb/doit">
<INPUT TYPE=PASSWORD NAME="passwd">
</FORM>
```

### See Also

- [TextField class](#)
- [FormElement class](#)

## PasswordField constructor

### Class

oracle.html.PasswordField

### Syntax

```
public PasswordField(String name)
```

```
public PasswordField(  
    String      name,  
    int         maxlength,  
    int         size,  
    String      value)
```

### Purpose

Creates a PasswordFiled object (with the specified attributes, if given).

### Parameters

name - name of this PasswordField object

maxLength - specifies the number of characters that can be entered into a text input field

size - specifies the amount of display space allocated to this input field

value - the initial value of the field

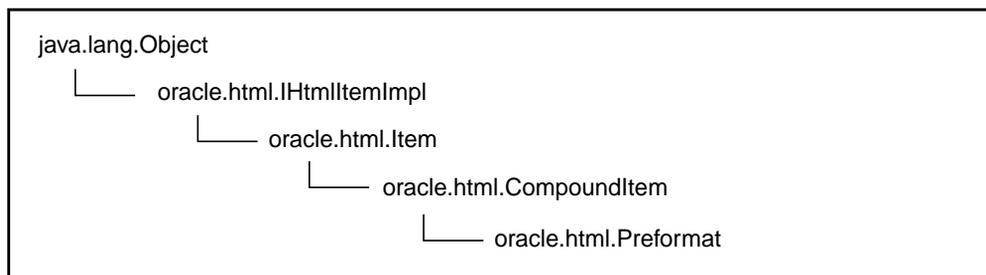
---

## Preformat class

public class **Preformat**  
extends `CompoundItem`

Preformatted text (PRE) as specified in HTML.

**Figure 1–45** Inheritance for the *Preformat* class



### Instructions

Use `Preformat` class for text that has been formatted for a monospaced font.

---

---

**Note:** Images and elements that produce changes in font size are illegal within a `Preformat` block.

---

---

### Example

```
// Create a Preformat object
Preformat pre = new Preformat();
// Add some preformatted text to this object
pre.addItem("      Name: YOUR NAME");
pre.addItem("      ID: YOUR ID");
```

### See Also

- [CompoundItem](#) class

## Preformat constructor

### Class

oracle.html.Preformat

### Syntax

```
public Preformat()
```

```
public Preformat(int width)
```

### Purpose

Creates a Preformat object (with specified attributes, if given).

### Parameters

width - the maximum number of characters for a line

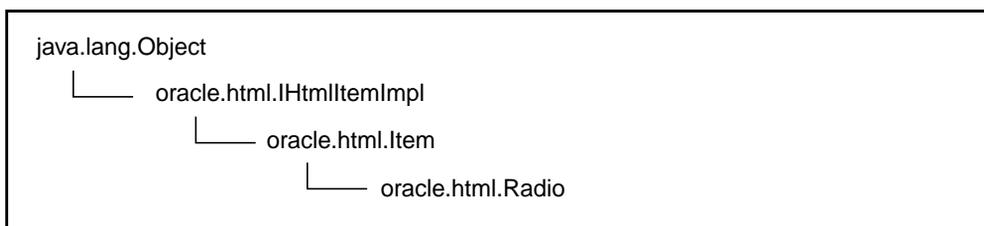
---

## Radio class

```
public class Radio  
extends Item
```

Radio encapsulates a Radio Button Element as specified in HTML2.

**Figure 1-46 Inheritance for the Radio class**



### Instructions

Use this class to present a list of mutually exclusive choices to the user. Usually a set of choices with the same name represents a 1-of-many choices field.

### Example

```
// Creates a Radio object (initial state == on)  
Radio radio = new Radio("flavor", "vanilla", true);
```

### See Also

- [Item class](#)

## Radio constructor

### Class

`oracle.html.Radio`

### Syntax

```
public Radio(  
    String      name,  
    String      value)
```

```
public Radio(  
    String      name,  
    String      value,  
    boolean     state)
```

### Purpose

Creates a Radio object with the specified attributes.

### Parameters

name - name of this Radio object

value - value of this object (text to appear in the browser)

state - default state (on/off)

## startTag

### Class

oracle.html.Radio

### Syntax

```
public String startTag()
```

### Returns

The start tag of this object.

## toHTML

### Class

oracle.html.Radio

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as an HTML string.

### Overrides

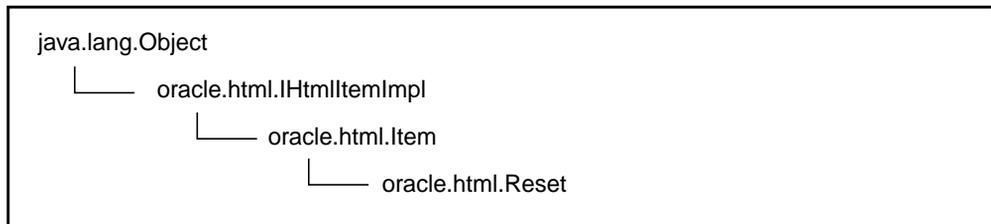
[toHTML](#) in [Item class](#).

## Reset class

```
public class Reset  
extends Item
```

The Reset class encapsulates an INPUT element with TYPE=RESET in HTML2. This element typically represents an input option, typically a button, that instructs the user agent to reset the form's fields to their initial states. The VALUE attribute, if present, indicates a label for the input (button).

**Figure 1–47 Inheritance for the Reset class**



### Instructions

Use this class to present the user with a button that, when pressed, will reset all values in the form to their initial states.

### Example

```
// Creates a Reset object  
Reset res = new Reset();
```

## Reset constructor

### Syntax

```
public Reset()  
  
public Reset(String value)
```

**Purpose**

Creates a Reset object.

**Parameters**

value - specifies the label of the Reset button

**startTag****Class**

oracle.html.Reset

**Syntax**

```
public String startTag()
```

**Returns**

The start tag of this object.

**toHTML****Class**

oracle.html.Reset

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this object as an HTML string.

## Overrides

[toHTML](#) in [Item](#) class.

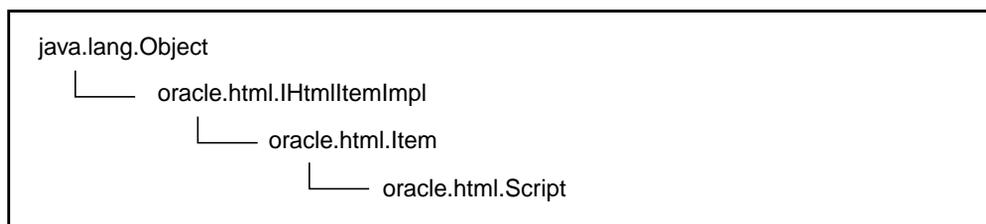
---

## Script class

public class **Script**  
extends Item

The Script class encapsulates a Script (SCRIPT) element.

**Figure 1–48 Inheritance for the Script class**



### Instructions

Use this class whenever you need to embed client-side script (e.g. Java Script or VBScript) in your HTML document to add interactivity to your document.

### Example

```
// Creates a Script object
Script s = new Script("VBScript");
// Adds some code to it
s.setCode("Document.write(\"Hello, World.\");");
```

### In HTML

```
<SCRIPT LANGUAGE="VBScript">Document.write("Hello, World.");</SCRIPT>
```

### See Also

- [IHtmlItem interface](#)
- [Item class](#)

## Script constructor

### Class

oracle.html.Script

### Syntax

```
public Script()
```

```
public Script(String language)
```

```
public Script(  
    String      language,  
    String      runAt)
```

### Purpose

Creates a Script object (with the specified language attribute, if given).

### Parameters

language - specifies the scripting language in which the scripting code is written. Possible values are VBScript and JScript

runat - specifies whether this is server side script or not. You can specify "SERVER" if this is server side script

## setCode

### Class

oracle.html.Script

### Syntax

```
public Script setCode(String code)
```

**Purpose**

Associate scripting code with this object.

**Parameters**

code - the script you want to associate with this object

**Returns**

`this`

**toHTML****Class**

`oracle.html.Script`

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this object as HTML string.

**Overrides**

[toHTML](#) in [Item](#) class.

**toString****Class**

`oracle.html.Script`

**Syntax**

```
public String toString()
```

### Returns

A string representation of this object.

### Overrides

[toString](#) in [Item](#) class.

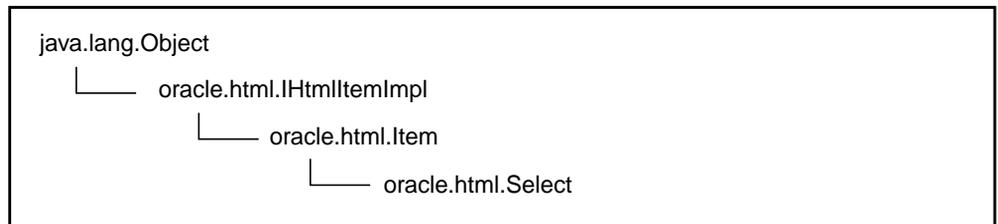
---

## Select class

public class **Select**  
extends Item

The Select class encapsulates the SELECT element in HTML2. This element constrains a form field to an enumerated list of values. The values are given in OPTION elements.

**Figure 1-49 Inheritance for the Select class**



### Instructions

Use this class in conjunction with the Option class to present the user an enumerated list of values.

### Example

```
// Creates a Select object with 5 choices. (frm is a Form object)
frm
    .addItem(new SimpleItem("Which server do you use?").setBold())
    .addItem(new Select("server")
        .addOption(new Option("Oracle WebServer"))
        .addOption(new Option("Netscape"))
        .addOption(new Option("Apache"))
        .addOption(new Option("Mosaic"))
        .addOption(new Option("CERN")))
```

### See Also

- [Option class](#)
- [FormElement class](#)

## Select constructor

### Syntax

```
public Select(String name)
```

```
public Select(  
    String      name,  
    int         size,  
    boolean     multiple)
```

### Purpose

Creates a Select object with the specified property name (and other attributes, if specified).

### Parameters

name - specifies a property name that is used to identify the user's choice when the form is submitted to the server

size - specifies the number of visible choices

multiple - specifies whether multiple selections are allowed

## addOption

### Syntax

```
public Select addOption(Option op)
```

### Purpose

Adds an option to the selection list.

### Parameters

op - a new option to be added to the Selection list

**Returns**

this

**See Also**

- [Option class](#)

## NumOptions

**Syntax**

```
public int NumOptions()
```

---

---

**Note:** This method is now obsolete and will be replaced by method [size\(\)](#).

---

---

**Returns**

The number of options present in this select object.

**See Also**

- [size](#)

## size

**Syntax**

```
public int size()
```

**Returns**

The number of options present in this object.

## toHTML

### Class

oracle.html.Script

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.Script

### Syntax

```
public String toString()
```

### Returns

A string representation of this object.

### Overrides

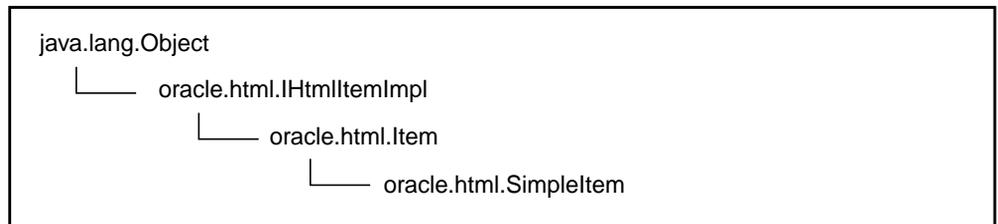
[toString](#) in [Item](#) class.

## SimpleItem class

public class **SimpleItem**  
extends Item

The SimpleItem class provides an object wrapper for any simple (Text String based) HTML Element. Since this class derives from Item class, most phrase markup capability is built-in.

*Figure 1–50 Inheritance for the SimpleItem class*



### Constants

Name	Syntax	Description
<b>HorizontalRule</b>	public static final SimpleItem HorizontalRule	HorizontalRule/HR element
<b>LineBreak</b>	public static final SimpleItem LineBreak	LineBreak/BR element
<b>Paragraph</b>	public static final SimpleItem Paragraph	Paragraph/P element

### Instructions

Use this class whenever you need to put a string with markup in your page.

### Example

```
// Creates a SimpleItem with String "HelloWorld!" (and BOLD it)
SimpleItem s = new SimpleItem("HelloWorld!").setBold();
```

### See Also

- [Item class](#)

## SimpleItem constructor

### Class

oracle.html.SimpleItem

### Syntax

```
public SimpleItem()
```

```
public SimpleItem(String s)
```

```
public SimpleItem(int i)
```

```
public SimpleItem(long l)
```

```
public SimpleItem(float f)
```

```
public SimpleItem(double d)
```

```
public SimpleItem(boolean b)
```

```
public SimpleItem(char c)
```

```
public SimpleItem(char data[])
```

```
public SimpleItem(Object o)
```

### Purpose

Creates an HTML SimpleItem.

## toHTML

### Class

oracle.html.SimpleItem

**Syntax**

```
public String toHTML()
```

**Returns**

The content of this item as a HTML string.

**Overrides**

[toHTML](#) in [Item class](#).

## toString

### Class

oracle.html.SimpleItem

### Syntax

```
public String toString()
```

### Returns

A string representation of this item.

### Overrides

[toString](#) in [Item](#) class.

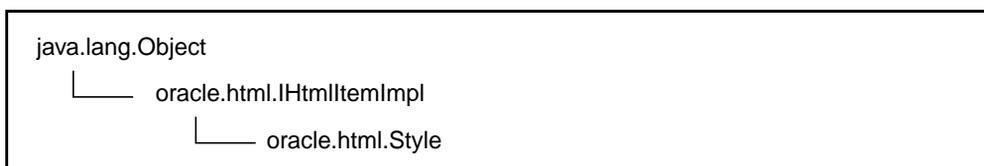
---

## Style class

**public class Style**  
extends IHtmlItemImpl

The Style and StyleElement class work together to provide embedding and inlining support for CSS1 (Cascading Style Sheet Level 1).

**Figure 1–51 Inheritance for the Style class**



### Instructions

Use instances of this class to group together stylistic properties (e.g. font, color, size, etc.) to define a particular style. Then associate instances of this class with instances of StyleElement class to associate a particular HTML element/tag with the defined style.

### Example

```
// Creates a new HtmlHead object
Htmlhead hd = new HtmlHead("Style Example");

// Creates a new Style object
Style s = new Style();
// Creates a new StyleElement object
StyleElement se = new StyleElement();

// adds a font property to this object
se.addProperty("font", "10pt Arial");

// adds the new object to the Style object
s.addElement("BODY", se);

// adds the style object to the head object
hd.addStyle(s);
```

### In HTML

```
<STYLE TYPE="text/css">
<!--
BODY {font : 10pt Arial }
-->
</STYLE>
```

### See Also

- [StyleElement class](#)
- [HtmlHead class](#)

## Style constructor

### Class

oracle.html.Style

### Syntax

```
public Style()
```

### Purpose

Creates an empty Style object.

## addElement

### Class

oracle.html.Style

### Syntax

```
public synchronized Style addElement(
```

```
String      tag,  
StyleElement elem)
```

### Purpose

Adds a style element object to the STYLE section, associate the element object with the specified HTML tag.

---

---

**Note:** Each tag in a STYLE section must be unique.

---

---

### Parameters

tag - the HTML tag whose style will be defined

elem - the StyleElement object which defines the style for the specified tag

### Returns

this

## setElementAt, elementAt, removeElementAt

### Class

oracle.html.Style

### Syntax

```
public synchronized Style setElementAt(  
    StyleElement elem,  
    int index)
```

```
public synchronized Style setElementAt(  
    StyleElement elem,  
    String tag)
```

```
public synchronized StyleElement elementAt(int index)
```

```
public synchronized StyleElement elementAt(String tag)
```

```
public synchronized StyleElement removeElementAt(int index)
```

```
public synchronized StyleElement removeElementAt(String tag)
```

### **Purpose**

`setElementAt` sets an style element at position given by `index`. The pervious style element at that position is discarded.

`elementAt` retrieves a `StyleElement` either at a specified position or with the specified HTML tag.

`removeElementAt` removes the element at the specified position or associated with the specified HTML tag and returns it.

### **Parameters**

`elem` - what the style element is to be set to

`index` - the specified position

`tag` - the HTML tag

### **Returns**

`setElementAt` returns `this`. `elementAt` returns the `StyleElement` at the specified position or with the specified HTML tag. `removeElementAt` returns the removed element.

## **elements**

### **Class**

`oracle.html.Style`

### **Syntax**

```
public synchronized Enumeration elements()
```

## positionOf

### Class

oracle.html.Style

### Syntax

```
public synchronized int positionOf(String tag)
```

### Purpose

Retrieves the position associated with the specified HTML tag.

### Parameters

tag - the specified HTML tag

### Returns

The position of the specified HTML tag.

## size

### Class

oracle.html.Style

### Syntax

```
public synchronized int size()
```

### Returns

The number of StyleElements in this object.

## toHTML

### Class

oracle.html.Style

### Syntax

```
public synchronized String toHTML()
```

### Returns

The content of the object as a HTML string.

### Overrides

[toHTML](#) in class [IHtmlItemImpl](#) class.

## toString

### Class

oracle.html.Style

### Syntax

```
public synchronized String toString()
```

### Overrides

[toString](#) in [Object](#) class.

---

## StyleElement class

**public class StyleElement**  
extends IHtmlItemImpl

The StyleElement and Style class work together to provide embedding and inlining support for CSS1 (Cascading Style Sheet Level 1).

**Figure 1–52 Inheritance for the StyleElement class**



### Instructions

Use instances of this class to group together stylistic properties (e.g. font, color, size, etc.) to define a particular style. Then associate instances of this class with instances of Style class to associate a particular HTML element/tag with the defined style.

### Example

```
// Creates a new HtmlHead object
Htmlhead hd = new HtmlHead("Style Example");

// Creates a new Style object
Style s = new Style();
// Creates a new StyleElement object
StyleElement se = new StyleElement();

// adds a font property to this object
se.addProperty("font", "10pt Arial");

// adds the new object to the Style object
s.addElement("BODY", se);

// adds the style object to the head object
hd.addStyle(s);
```

### See Also

- [Style class](#)
- [HtmlHead class](#)

## StyleElement constructor

### Class

oracle.html.StyleElement

### Syntax

```
public StyleElement()
```

### Purpose

Creates an empty StyleElement object.

## addProperty

### Class

oracle.html.StyleElement

### Syntax

```
public synchronized StyleElement addProperty(  
    String      name,  
    String      value)
```

### Purpose

Adds a style property (name-value pair) to this StyleElement object.

**Parameters**

name - name of the style (e.g. "font")

value - value of the specified style (e.g. "10pt")

**Returns**

this

**properties****Class**

oracle.html.StyleElement

**Syntax**

```
public synchronized Enumeration properties()
```

**setPropertyAt, propertyAt****Class**

oracle.html.StyleElement

**Syntax**

```
public synchronized StyleElement setPropertyAt(  
    String      name,  
    String      value,  
    int         index)
```

```
public synchronized String propertyAt(int index)
```

**Purpose**

Sets or retrieves a property at the position given by index.

---

---

**Note:** When setting properties, the property formerly at that position will be discarded.

---

---

### Parameters

name - the name of the property (e.g. "font")

value - the value of the property (e.g. "10pt Arial")

index - the specified position

### Returns

setPropertyAt returns `this`. propertyAt returns the property at the specified position.

## size

### Class

oracle.html.StyleElement

### Syntax

```
public synchronized int size()
```

### Returns

The number of properties in this object.

## toHTML

### Class

oracle.html.StyleElement

**Syntax**

```
public synchronized String toHTML()
```

**Returns**

The content of object as HTML string.

**Overrides**

[toHTML](#) in [IHtmlItemImpl](#) class.

**toString****Class**

oracle.html.StyleElement

**Syntax**

```
public synchronized String toString()
```

**Overrides**

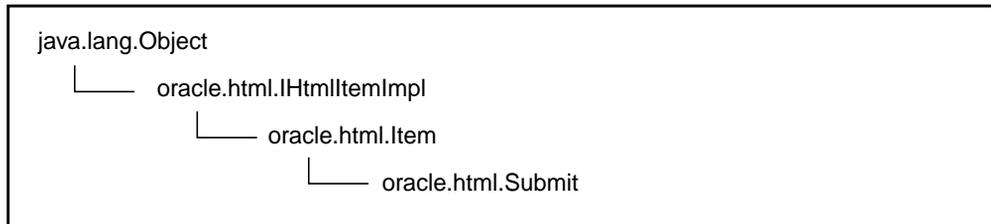
[toString](#) in [Object](#) class.

## Submit class

```
public class Submit  
extends Item
```

The Submit class encapsulates the INPUT element with type=SUBMIT in HTML2. This element represents an input option, typically a button, that instructs the user agent to submit the form.

*Figure 1–53 Inheritance for the Submit class*



### Instructions

Use this class to add a submission button to the form in your HTML page.

### Example

```
// Creates a Submit Button  
Submit submit = new Submit();  
// Creates a Submit Button which constitutes a form field and has a value  
// (the value is what shows up as text of the button)  
Submit submit = new Submit("Submit_Name", "Submit!");
```

### See Also

- [Form class](#)
- [Item class](#)

## Submit constructor

### Class

oracle.html.Submit

### Syntax

```
public Submit()
```

```
public Submit(  
    String      name,  
    String      value)
```

### Purpose

Creates a Submit object.

### Parameters

name - specifies the input field value of this submit button

value - specifies the button's label

## toHTML

### Class

oracle.html.Submit

### Syntax

```
public String toHTML()
```

### Returns

The content of this object as HTML string.

## Overrides

[toHTML](#) in [Item](#) class.

---

## Table class

public abstract class **Table**  
extends Item  
implements IAlign, ITableFrame, ITableRules

The Table class is an abstract base class that provides the foundation for all Table (Dynamic/Fixed) classes.

---

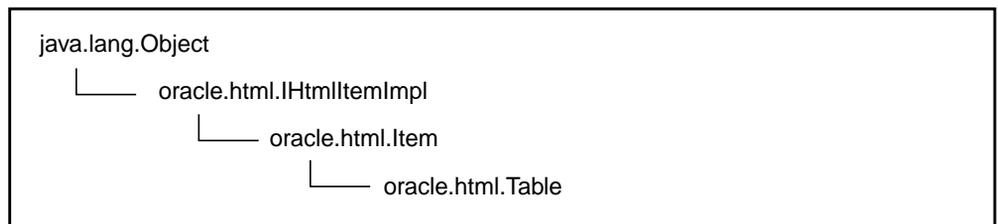
---

**Note:** You cannot directly instantiate a Table object. To create a table in HTML, you need to instantiate either a DynamicTable object or a StaticTable object (not yet implemented).

---

---

*Figure 1-54 Inheritance for the Table class*



### Instructions

Since Table is an abstract class, you cannot instantiate it directly.

### Example

To instantiate a Table object, either derive your own class from Table or instantiate from DynamicTable class (recommended).

### See Also

- [DynamicTable class](#)
- [TableRow class](#)
- [TableRowCell class](#)
- [TableDataCell class](#)

- [TableHeaderCell class](#)

## Table constructor

### Class

oracle.html.Table

### Syntax

```
public Table()
```

```
public Table(  
    String      caption,  
    String      width,  
    int         border,  
    int         cellspacing,  
    int         cellpadding)
```

### Purpose

Creates a Table object.

### Parameters

caption - caption of the table

border - the width of border framing the table

cellspacing - the space between individual cells in a table

cellpadding - the amount of space between the border & content of cell

width - the desired width of the table (standard units or "%")

halign - the horizontal position of the table

frame - specifies which side of the frame to render

rules - specifies where to draw rules within the table interior

## setBackgroundcolor, backgroundColor

### Class

oracle.html.Table

### Syntax

```
public Table setBackgroundcolor(String color)
```

```
public Table setBackgroundcolor(Color color)
```

```
public String backgroundColor()
```

### Purpose

Sets or returns the background color of this table.

### Parameters

color - the background color of this table

## setBackgroundimage, backgroundImage

### Class

oracle.html.Table

### Syntax

```
public Table setBackgroundimage(String url)
```

```
public String backgroundImage()
```

### Purpose

Sets or returns the URL for the background image of this table.

## setBorder, border

### Class

oracle.html.Table

### Syntax

```
public Table setBorder(int border)
```

```
public int border()
```

### Purpose

Sets or returns the BORDER attribute of this table.

## setBorderColor, borderColor

### Class

oracle.html.Table

### Syntax

```
public Table setBorderColor(String color)
```

```
public Table setBorderColor(Color color)
```

```
public String borderColor()
```

### Purpose

Sets or returns the border color of this table.

## setCaption, caption

### Class

oracle.html.Table

### Syntax

```
public Table setCaption(String caption)
```

```
public Table setCaption(  
    String      caption,  
    String      align)
```

```
public String caption()
```

### Purpose

Sets or returns the caption of this table.

### Parameters

caption - the caption of this Table object

align - the alignment { TOP | BOTTOM } of this caption

### Returns

setCaption returns the caption of this table. caption returns *this*.

## captionAlign

### Class

oracle.html.Table

### Syntax

```
public String captionAlign()
```

### Returns

The caption alignment of this table.

## setCellPadding, cellPadding

### Class

oracle.html.Table

### Syntax

```
public Table setCellPadding(int cellpadding)
```

```
public int cellPadding()
```

### Purpose

Sets or returns the CELLPADDING attribute of this table.

## setCellSpacing, cellSpacing

### Class

oracle.html.Table

### Syntax

```
public Table setCellSpacing(int cellspacing)
```

```
public int cellspacing()
```

### Purpose

Sets or returns the CELLSPACING attribute of this table.

## startTag, endTag

### Class

oracle.html.Table

### Syntax

```
public String startTag()
```

```
public String endTag()
```

### Returns

Either the start tag or the end tag of this object.

## setFrame, frame

### Class

oracle.html.Table

### Syntax

```
public Table setFrame(int frame)
```

```
public int frame()
```

### Purpose

Sets or returns the FRAME attribute of this table.

## setHAlign, hAlign

### Class

oracle.html.Table

### **Syntax**

```
public Table setIAlign(int ha)

public int hAlign()
```

### **Purpose**

Sets or returns the HALIGN attribute of this object.

## **setRules, rules**

### **Class**

oracle.html.Table

### **Syntax**

```
public Table setRules(int rules)

public int rules()
```

### **Purpose**

Sets or returns the RULES attribute of this table.

## **setWidth, width**

### **Class**

oracle.html.Table

### **Syntax**

```
public Table setWidth(String width)

public String width()
```

**Purpose**

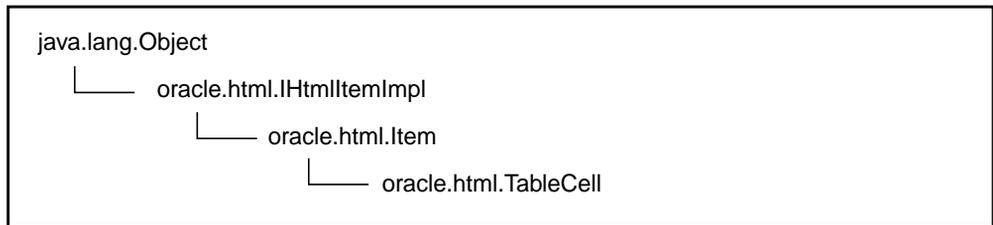
Sets or returns the WIDTH attribute of this table.

## TableCell class

**public abstract class TableCell**  
 extends Item  
 implements IAlign, IValign

TableCell is an abstract base class for all TableCells.

**Figure 1–55 Inheritance for the TableCell class**



### Instructions

You cannot directly instantiate a TableCell object. You usually use a TableCell object either by instantiating a TableCell object or a TableCell object.

### See Also

- [TableCell class](#)
- [TableRowCell class](#)
- [TableCell class](#)

## TableCell constructor

### Class

oracle.html.TableCell

### Syntax

public TableCell()

## setBackgroundcolor, backgroundColor

### Class

oracle.html.TableCell

### Syntax

```
public TableCell setBackgroundcolor(String color)
```

```
public TableCell setBackgroundcolor(Color color)
```

```
public String backgroundColor()
```

### Purpose

Sets or returns the background color of this cell.

## setHAlign, halign

### Class

oracle.html.TableCell

### Syntax

```
public TableCell setHAlign(int halign)
```

```
public int halign()
```

### Purpose

Sets or returns the horizontal alignment attribute for this cell.

## setBorderColor

### Class

oracle.html.TableCell

### Syntax

```
public TableCell setBorderColor(String color)
```

```
public TableCell setBorderColor(Color color)
```

### Purpose

Sets the border color for this cell.

## setVAlign, valign

### Class

oracle.html.TableCell

### Syntax

```
public TableCell setVAlign(int valign)
```

```
public int valign()
```

### Purpose

Sets or returns the vertical alignment attribute for this cell.

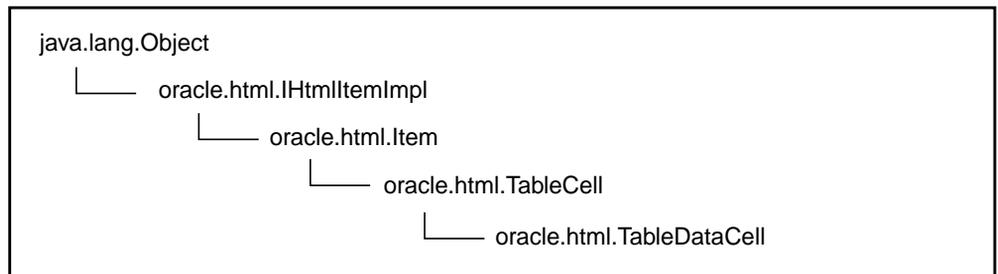
---

## TableDataCell class

```
public class TableDataCell  
extends TableCell
```

TableDataCell encapsulates a TD (table data cell) element.

**Figure 1–56 Inheritance for the TableDataCell class**



### Instructions

Use this class whenever you want to construct a TD element within a DynamicTable.

### Example

```
// Create a TableDataCell with a simple string as content  
TableDataCell cell = new TableDataCell("Simple Cell");
```

### See Also

- [TableCell class](#)
- [TableHeaderCell class](#)
- [DynamicTable class](#)

## TableDataCell constructor

### Class

oracle.html.TableDataCell

### Syntax

```
public TableDataCell()  
  
public TableDataCell(String content)  
  
public TableDataCell(IHtmlItem content)  
  
public TableDataCell(  
    int         valign,  
    int         halign,  
    int         rowspan,  
    int         colspan,  
    boolean     nowrap,  
    IHtmlItem   content)  
  
public TableDataCell(  
    int         valign,  
    int         halign,  
    int         height,  
    int         width,  
    int         rowspan,  
    int         colspan,  
    boolean     nowrap,  
    IHtmlItem   content)
```

### Purpose

Creates a TableDataCell object.

### Parameters

valign - vertical alignment

halign - horizontal alignment

height - suggested height of this cell in pixels

width - suggested width of this cell in pixels

rowspan - number of rows spanned by cell

colspan - number of columns spanned by cell

nowrap - specifies whether to suppress word wrap in TH or TD cell

content - content of this cell

## addItem

### Class

oracle.html.TableDataCell

### Syntax

```
public TableDataCell addItem(String str)
```

```
public TableDataCell addItem(IHtmlItem newItem)
```

### Purpose

Adds either a string or an HTML item to the content of this cell.

### Parameters

str - the string to be added to the content of this cell

newItem - the new object to be added to the content of this cell

### Returns

this

## setColSpan, colSpan

### Class

oracle.html.TableDataCell

### Syntax

```
public TableDataCell setColSpan(int colSpan)
```

```
public int colSpan()
```

### Purpose

Sets or returns the value of COLSPAN attribute for this object.

## setHeight, height

### Class

oracle.html.TableDataCell

### Syntax

```
public TableDataCell setHeight(int height)
```

```
public int height()
```

### Purpose

Sets or returns the height attribute of this object.

### Parameters

height - the height of this cell in pixels

## setNoWrap, noWrap

### Class

oracle.html.TableDataClass

### Syntax

```
public TableDataCell setNoWrap(boolean noWrap)
```

```
public boolean noWrap()
```

### Purpose

Sets or returns the NOWRAP attribute of this object.

## setRowSpan, rowSpan

### Class

oracle.html.TableDataClass

### Syntax

```
public TableDataCell setRowSpan(int rowSpan)
```

```
public int rowSpan()
```

### Purpose

Sets or returns the ROWSPAN attribute of this object.

## setWidth, width

### Class

oracle.html.TableDataClass

### Syntax

```
public TableCell setWidth(int width)
```

```
public int width()
```

### Purpose

Sets or returns the width attribute of this object.

### Parameters

width - suggested width of this cell in pixels

## toHTML

### Class

oracle.html.TableDataClass

### Syntax

```
public String toHTML()
```

### Returns

The content of this item as a HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.TableDataClass

## Syntax

```
public String toString()
```

---

---

**Note:** `toString` assumes that the cell contains only strings as its content.

---

---

## Returns

A string representation of this item.

## Overrides

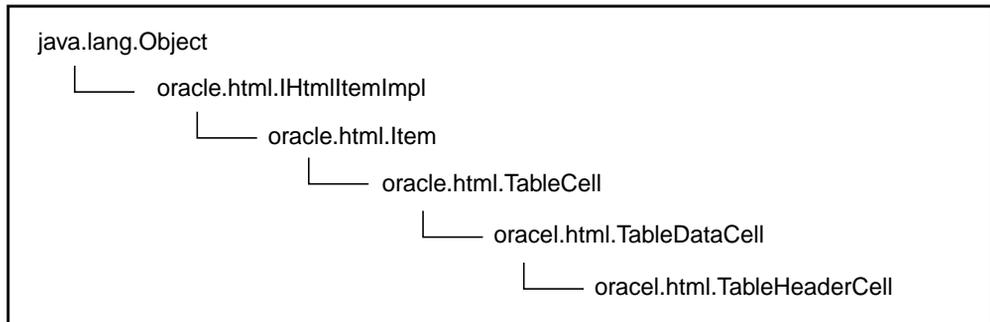
[toString](#) in [Item](#) class.

## TableHeaderCell class

```
public class TableHeaderCell  
extends TableCell
```

TableHeaderCell encapsulates a TH element.

**Figure 1–57 Inheritance for the TableHeaderCell class**



### Instructions

Use this class whenever you want to use a header element in a Dynamic Table.

### Example

```
// Create a TableHeaderCell with a simple string as content  
TableHeaderCell cell = new TableHeaderCell("Simple Header Cell");
```

### See Also

- [TableCell class](#)
- [TableDataCell class](#)
- [DynamicTable class](#)

## TableHeaderCell constructor

### Class

oracle.html.TableHeaderCell

### Syntax

```
public TableHeaderCell(String content)
```

```
public TableHeaderCell(IHtmlItem content)
```

```
public TableHeaderCell(  
    int          valign,  
    int          halign,  
    int          rowspan,  
    int          colspan,  
    boolean      noWrap,  
    IHtmlItem    content)
```

### Purpose

Creates a TableHeaderCell object.

### Parameters

valign - vertical alignment

halign - horizontal alignment

rowspan - number of rows spanned by cell

colspan - number of columns spanned by cell

nowrap - specifies whether to suppress word wrap in TH or TD cell

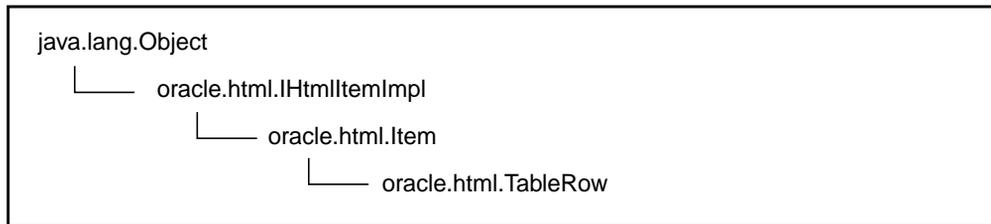
content - content of this cell

## TableRow class

```
public class TableRow  
extends Item
```

The TableRow class abstracts a row of a table.

**Figure 1–58** Inheritance for the TableRow class



### Instructions

Use this class together with Table[Data/Header]Cell, DynamicTable classes to create an HTML table.

### Example

```
// Creates a new TableRow  
TableRow row = new TableRow;  
// Add TableDataCells to TableRow  
row  
    .addCell(new TableDataCell(new SimpleItem("first cell")))  
    .addCell(new TableDataCell(new SimpleItem("second cell")))  
    .addCell(new TableDataCell(new SimpleItem("third cell")));
```

### See Also

- [Table class](#)
- [DynamicTable class](#)
- [TableHeaderCell class](#)
- [TableDataCell class](#)

## TableRow constructor

### Class

oracle.html.TableRow

### Syntax

```
public TableRow()  
  
public TableRow(int initCapacity)  
  
public TableRow(  
    int          initCapacity,  
    int          halign,  
    int          valign)
```

### Purpose

Creates a new TableRow object.

### Parameters

initCapacity - the initial capacity of the new row

halign - the horizontal alignment

valign - the horizontal alignment

## addCell

### Class

oracle.html.TableRow

### Syntax

```
public synchronized TableRow addCell(String str)  
  
public synchronized TableRow addCell(TableCell cell)
```

### **Purpose**

Adds a cell to this TableRow object.

## **setCellAt, cellAt**

### **Class**

oracle.html.TableRow

### **Syntax**

```
public synchronized TableRow setCellAt(  
    int          index,  
    SimpleItem  item)
```

```
public synchronized TableCell cellAt(int index)
```

### **Purpose**

Sets or returns the TableCell attribute of this object at a specified position.

## **cells**

### **Class**

oracle.html.TableRow

### **Syntax**

```
public int cells()
```

### **Returns**

The number of cells in this TableRow object.

## setBackgroundcolor

### Class

oracle.html.TableRow

### Syntax

```
public TableRow setBackgroundcolor(String color)
```

```
public TableRow setBackgroundcolor(Color color)
```

### Purpose

Sets the background color of this table row.

## setbordercolor

### Class

oracle.html.TableRow

### Syntax

```
public TableRow setbordercolor(String color)
```

```
public TableRow setbordercolor(Color color)
```

### Purpose

Sets the border color for this TableRow object.

## setHAlign, setVAlign

### Class

oracle.html.TableRow

### Syntax

```
public TableRow setHAlign(int halign)
```

```
public TableRow setVAlign(int valign)
```

### Purpose

Sets the default horizontal or vertical alignment attribute for all cells in this row.

## toHTML

### Class

oracle.html.TableRow

### Syntax

```
public synchronized String toHTML()
```

### Returns

The content of this item as a HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.TableRow

### Syntax

```
public synchronized String toString()
```

### Returns

A string representation of this item.

### Overrides

[toString](#) in [Item](#) class.

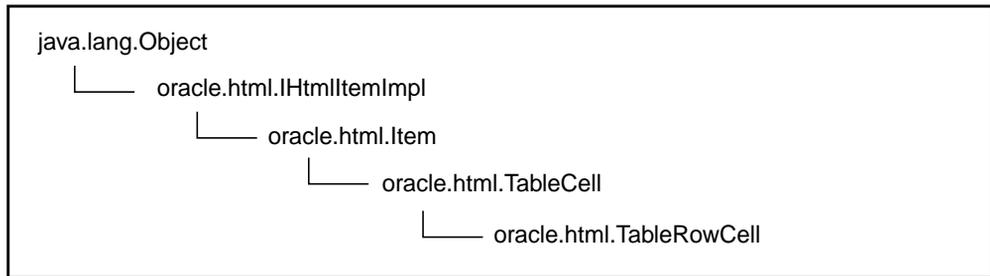
## TableRowCell class

---

```
public class TableRowCell  
extends TableCell
```

TableRowCell encapsulates a TR element.

**Figure 1–59** *Inheritance for the TableRowCell class*



### Instructions

This class is normally wrapped in TableRow and automatically instantiated when you need it.

### Example

You don't normally instantiate a TableRowCell. Although you can do it:

```
// Instantiate a TableRowCell object  
TableRowCell cell = new TableRowCell();
```

### See Also

- [TableCell class](#)
- [TableRow class](#)

## TableRowCell constructor

### Class

oracle.html.TableRowCell

### Syntax

```
public TableRowCell()  
  
public TableRowCell(  
    int          halign,  
    int          valign)
```

### Purpose

Creates a TableRowCell object with default attributes.

## toHTML

### Class

oracle.html.TableRowCell

### Syntax

```
public String toHTML()
```

### Returns

The content of this item as a HTML string.

### Overrides

[toHTML](#) in [Item class](#).

## toString

### Class

oracle.html.TableRowCell

### Syntax

```
public String toString()
```

### Returns

A string representation of this item.

### Overrides

[toString](#) in [Item](#) class.

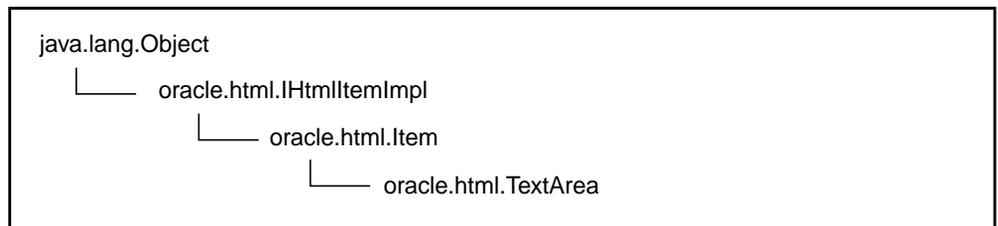
---

## TextArea class

public class **TextArea**  
extends Item

TextArea class encapsulates the TEXTAREA input element in HTML2. The TextArea element represents a multi-line text field.

**Figure 1–60 Inheritance for the TextArea class**



### Instructions

Use this class to present a multi-line text field to the user for input purposes.

### Example

```
// Creates a text area object with 40 columns and 10 rows
TextArea ta = new TextArea("comments", 40, 10);
```

### See Also

- [Item class](#)
- [TextField class](#)
- [FormElement class](#)
- [Form class](#)

## TextArea constructor

### Class

oracle.html.TextArea

### Syntax

```
public TextArea(String name,  
               int      cols,  
               int      rows)
```

```
public TextArea(String name,  
               int      cols,  
               int      rows,  
               String   val)
```

### Purpose

Creates a TextArea object.

### Parameters

name - specifies a property name that is used to identify the textarea field when the form is submitted to server

cols - specifies the visible width in average character width

rows - specifies the number of visible text lines

val - the default content of this element

## setCols, setRows

### Class

oracle.html.TextArea

**Syntax**

```
public TextArea setCols(int cols)
```

```
public TextArea setRows(int rows)
```

**Purpose**

Sets either the ROWS or COLS attribute of this object.

**Parameters**

cols - specifies the visible width in average character width

rows - specifies the number of visible text lines

**Returns**

```
this
```

**setDefaultContent****Class**

```
oracle.html.TextArea
```

**Syntax**

```
public TextArea setDefaultContent(String val)
```

**Purpose**

Sets this object's initial value / default content.

**Parameters**

val - the default content of this element

**Returns**

```
this
```

## toHTML

### Class

oracle.html.TextArea

### Syntax

```
public String toHTML()
```

### Returns

The content of this item as a HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

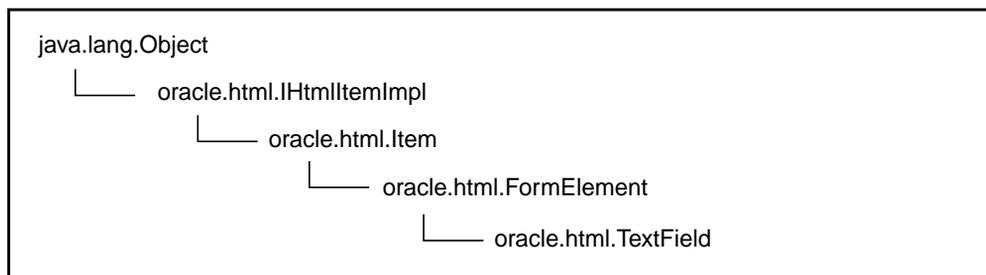
---

## TextField class

```
public class TextField
extends FormElement
```

TextField class encapsulates a single line TextField element.

**Figure 1–61** Inheritance for the TextField class



### Instructions

Use this class to add a single line text input field on your HTML document.

### Example

```
// Create a TextField object
TextField tf = new TextField("Name");
```

### See Also

- [TextArea class](#)
- [Form class](#)
- [FormElement class](#)

## TextField constructor

### Class

oracle.html.TextField

### Syntax

```
public TextField(String name)
```

```
public TextField(  
    String      name,  
    int         maxLength,  
    int         size,  
    String      value)
```

### Purpose

Creates a TextField object.

### Parameters

name - specifies name of this TextField object

maxLength - specifies the number of characters that can be entered into a text input field. If the value of maxLength is greater than the value of the size attribute, the field should scroll appropriately. The default number of characters is unlimited.

size - specifies the amount of display space allocated to this input field according to its type. The default depends on the user agent.

value - the default value of this field

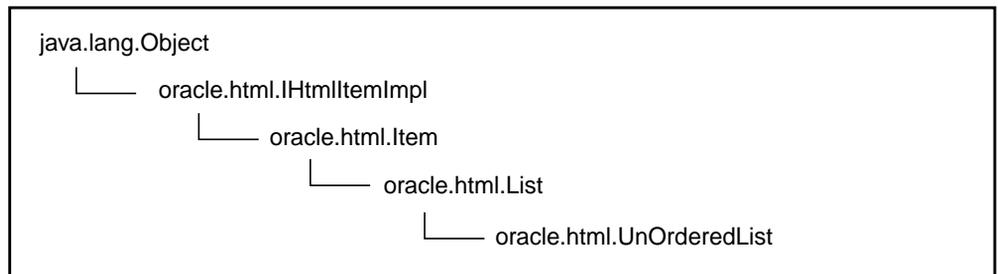
---

## UnOrderedList class

```
public class UnOrderedList  
extends List
```

UnOrderedList encapsulates an Unordered (UL) list element in HTML.

**Figure 1–62** *Inheritance for the UnOrderedList class*



### Instructions

Use this class whenever you need to present the user with a list of items (without order).

### Example

```
// Create an UnOrdered List  
UnOrderedList ul = new UnOrderedList();  
ul.addItem(new SimpleItem("UnOrdered1"));  
ul.addItem(new SimpleItem("UnOrdered2"));  
ul.addItem(new SimpleItem("UnOrdered3"));
```

### See Also

- [Item class](#)
- [List class](#)

## UnOrderedList constructor

### Class

oracle.html.UnOrderedList

### Syntax

```
public UnOrderedList()
```

```
public UnOrderedList(  
    String          type,  
    boolean         compact)
```

### Purpose

Creates an UnOrderedList object.

### Parameters

type - specifies bullet style

compact - specifies whether inter-item spacing should be reduced

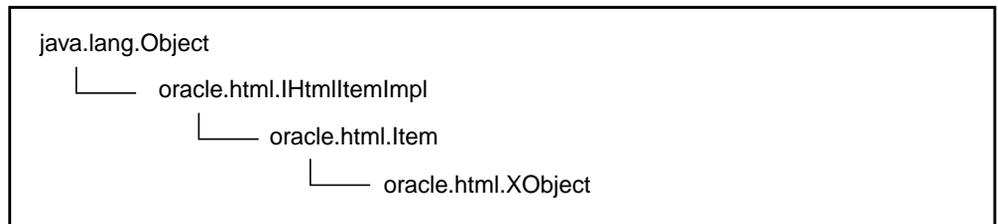
---

## XObject class

```
public class XObject
    extends Item
    implements IAlign, IAlign
```

XObject class encapsulates an OBJECT element as specified in WWW Consortium (W3C) object model extension to HTML 3.2.

**Figure 1–63 Inheritance for the XObject class**



### Instructions

Use this class to insert an arbitrary object into your HTML document. An object can be an ActiveX control, an image, or even another document.

### Example

```
// Create an XObject
XObject obj = new XObject("clsid:663C8FEF-1EF9-11CF-A3D8-080036F12502",
    null, null, "http://www.acme.com/clock.stm", false);
// Add parameters to this ActiveX control object
obj.addParam("TICKS", "100");
```

### In HTML

```
<Object CLASSID="clsid:663C8FEF-1EF9-11CF-A3D8-080036F12502"
DATA="http://www.acme.com/clock.stm">
<PARAM NAME="TICKS" VALUE="100">
</Object>
```

### See Also

- [Embed class](#)
- [Applet class](#)

## XObject constructor

### Class

oracle.html.XObject

### Syntax

```
public XObject(  
    String      classID,  
    String      codeBase,  
    String      codeType,  
    String      data,  
    boolean     declare)
```

```
public XObject(  
    String      classID,  
    String      codeBase,  
    String      codeType,  
    String      data,  
    boolean     declare,  
    String      align,  
    int         border,  
    int         width,  
    int         height,  
    int         vspace,  
    int         hspace,  
    String      name,  
    String      standby,  
    boolean     shapes,  
    String      useMap)
```

### Purpose

Creates an XObject with the specified attributes.

## Parameters

**classid** - identifies the implementation for the object. Syntax depends on the object type. For example,

- for ActiveX Controls, the syntax is: `CLSID::class-identifier`
- for Java Applets, the syntax is: `java:class name`

**codebase** - identifies the code base for the object Syntax depends on the object type

**codetype** - specifies the internet media type for code

**data** - specifies data for the object. Syntax depends on the object type

**declare** - specifies whether this is a declaration or an instantiation

**align** - specifies the alignment attribute of the object

**border** - specifies the width of the border

**width** - suggested width in pixels

**height** - suggested height in pixels

**vspace** - number of pixels above and below the object

**hspace** - number of pixels on each side of the object

**name** - sets the name of the object when submitted as part of a form

**standby** - specifies the message to show while loading the object

**shapes** - specifies whether the object has shaped hyperlinks

**usemap** - specifies the image map to use with the object

## addParam

### Class

`oracle.html.XObject`

### Syntax

```
public XObject addParam(  
    String      name,  
    String      value)
```

### **Purpose**

Adds a parameter (object specific attribute).

### **Parameters**

name - name of the attribute/parameter

value - value of the parameter

### **Returns**

this

## **setAlign**

### **Class**

oracle.html.XObject

### **Syntax**

```
public XObject setAlign(String align)
```

### **Purpose**

Sets the alignment attribute of the object.

### **Parameters**

align - specifies the alignment attribute of the object

### **Returns**

this

## setBorder

### Class

oracle.html.XObject

### Syntax

```
public XObject setBorder(int border)
```

### Purpose

Specifies the width of the border.

### Parameters

border - Specifies the width of the border

### Returns

this

## setClassID

### Class

oracle.html.XObject

### Syntax

```
public XObject setClassID(String classID)
```

### Purpose

Specifies the implementation for the object.

### Parameters

classid - identifies the implementation for the object. Syntax depends on the object type. For example,

- for ActiveX Controls, the syntax is: CLSID::class-identifier

- for Java Applets, the syntax is: `java: class name`

### Returns

`this`

## setCodeBase

### Class

`oracle.html.XObject`

### Syntax

```
public XObject setCodeBase(String codeBase)
```

### Purpose

Sets the code base (the base URL of the object).

### Parameters

`codebase` - identifies the code base for the object. Syntax depends on the object type

### Returns

`this`

## setCodeType

### Class

`oracle.html.XObject`

### Syntax

```
public XObject setCodeType(String codeType)
```

**Purpose**

Sets the internet media type for code.

**Parameters**

codeType - specifies the internet media type for code

**Returns**

this

**setData****Class**

oracle.html.XObject

**Syntax**

```
public XObject setData(String data)
```

**Purpose**

Sets the data for the object.

**Parameters**

data - specifies data for the object. Syntax depends on the object type

**Returns**

this

**setDeclare****Class**

oracle.html.XObject

### **Syntax**

```
public XObject setDeclare(boolean declare)
```

### **Purpose**

Specifies whether this is a declaration or an instantiation.

### **Parameters**

declare - specifies whether this is a declaration or an instantiation

### **Returns**

this

## **setHeight, setWidth**

### **Class**

oracle.html.XObject

### **Syntax**

```
public XObject setHeight(int height)
```

```
public XObject setWidth(int width)
```

### **Purpose**

Sets either the height or width of the object window in pixels.

### **Parameters**

height - suggested height in pixels

width - suggested width in pixels

### **Returns**

this

## setHSpace, setVSpace

### Class

oracle.html.XObject

### Syntax

```
public XObject setHSpace(int hspace)
```

```
public XObject setVSpace(int vspace)
```

### Purpose

setHSpace sets the number of pixels on each side of the object window.

setVSpace sets the number of pixels above and below the object window.

### Parameters

hspace - number of pixels on each side of the object

vspace - number of pixels above and below the object

### Returns

this

## setName

### Class

oracle.html.XObject

### Syntax

```
public XObject setName(String name)
```

### Purpose

Sets the name of the object when submitted as part of a form.

### Parameters

name - Sets the name of the object when submitted as part of a form

### Returns

this

## setShapes

### Class

oracle.html.XObject

### Syntax

```
public XObject setShapes(boolean shapes)
```

### Purpose

Specifies whether the object has shaped hyperlinks.

### Parameters

shapes - specifies whether the object has shaped hyperlinks

### Returns

this

## setStandby

### Class

oracle.html.XObject

### Syntax

```
public XObject setStandby(String standby)
```

**Purpose**

Specifies the message to show while loading the object.

**Parameters**

standby - specifies the message to show while loading the object

**Returns**

`this`

**setUseMap****Class**

`oracle.html.XObject`

**Syntax**

```
public XObject setUseMap(String useMap)
```

**Purpose**

Specifies the image map to use with the object.

**Parameters**

`useMap` - specifies the image map to use with the object

**Returns**

`this`

**toHTML****Class**

`oracle.html.XObject`

### Syntax

```
public String toHTML()
```

### Returns

The content of this item as a HTML string.

### Overrides

[toHTML](#) in [Item](#) class.

## toString

### Class

oracle.html.XObject

### Syntax

```
public String toString()
```

### Returns

A string representation of this item.

### Overrides

[toString](#) in [Item](#) class.

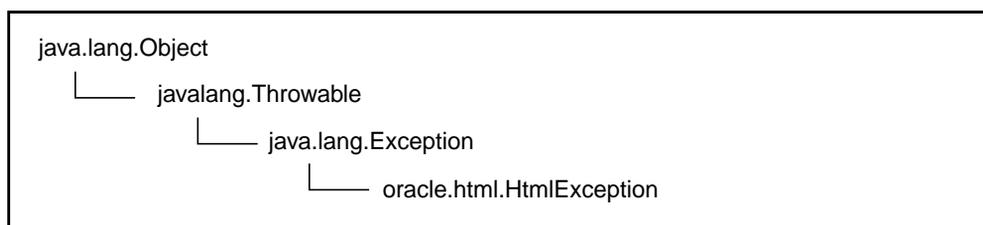
---

## HtmlException class

```
public class HtmlException  
extends Exception
```

Signals that an error condition has occurred.

**Figure 1–64** Inheritance for the *HtmlException* class



## HtmlException constructor

### Class

`oracle.html.HtmlException`

### Syntax

```
public HtmlException()
```

```
public HtmlException(String s)
```

### Purpose

Constructs an `HtmlException` (with the specified error message, if given).

### Parameters

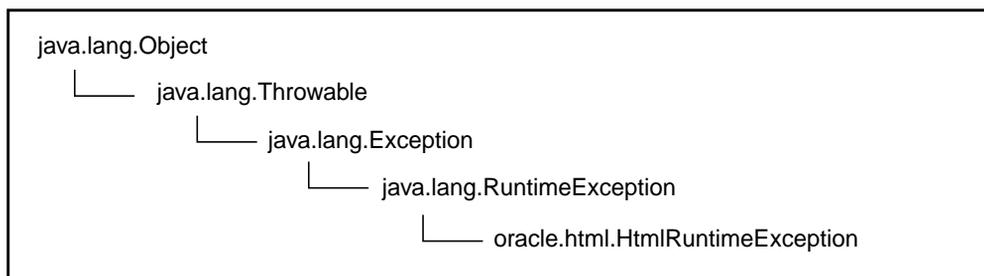
`msg` - the error message

## HtmlRuntimeException class

**public class HtmlRuntimeException**  
extends RuntimeException

Signals that an error condition has occurred.

*Figure 1–65 Inheritance for the HtmlRuntimeException class*



## HtmlRuntimeException constructor

### Class

oracle.html.HtmlRuntimeException

### Syntax

```
public HtmlRuntimeException()
```

```
public HtmlRuntimeException(String s)
```

### Purpose

Constructs an HtmlRuntimeExcption (with the specified error message, if given).

### Parameters

msg - the error message

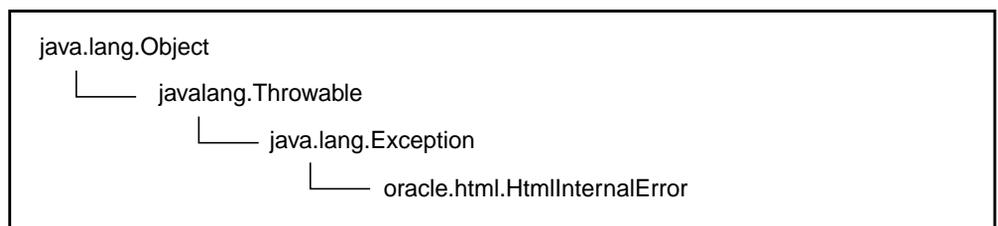
---

## HtmlInternalError class

```
public class HtmlInternalError  
extends Error
```

Signals that an internal error condition has occurred.

**Figure 1–66** *Inheritance for the HtmlInternalError class*



## HtmlInternalError constructor

### Class

oracle.html.HtmlInternalError

### Syntax

```
public HtmlInternalError()
```

```
public HtmlInternalError(String s)
```

### Purpose

Constructs an HtmlInternalError object (with the specified error message, if given).

### Parameters

msg - the error message



---

---

## oracle.OAS.servlet.http package

This chapter contains reference pages for the following classes.

- oracle.OAS.servlet.ServletRequest
  - [getSession](#)

---

---

**Note:** The `getSession` method is part of the `oracle.OAS.servlet.ServletRequest` package.

---

---

- [HttpSession](#) class
  - [getCreationTime](#)
  - [getId](#)
  - [getLastAccessedTime](#)
  - [getMaxInactiveInterval](#)
  - [getValue](#)
  - [getValueNames](#)
  - [invalidate](#)
  - [isNew](#)
  - [putValue](#)
  - [removeValue](#)
  - [setMaxInactiveInterval](#)

This chapter contains reference pages for the following exception classes:

---

---

**Note:** These exception classes are part of the `oracle.oas.session` package.

---

---

- [InactiveSessionException](#) class
- [InvalidNameException](#) class
- [NullSessionStateException](#) class
- [ObjectLookupFailureException](#) class
- [ObjectNotFoundException](#) class
- [ObjectUpdateFailureException](#) class
- [SessionDestroyFailureException](#) class
- [SessionException](#) class
- [SessionLockedException](#) class
- [SessionLookupFailureException](#) class
- [SessionNotFoundException](#) class

## getSession

### Class

`oracle.OAS.servlet.ServletRequest`

---

---

**Note:** This method is part of the `oracle.OAS.servlet.ServletRequest` package.

---

---

### Syntax

```
public static Session getSession(boolean create) throws Exception
```

```
public static Session getSession(  
    boolean create,
```

---

boolean            distributed,  
boolean            secure) throws Exception

## Purpose

Gets the current valid session associated with the HTTP request. If `create` is `true`, a new session will be created only if one does not currently exist. If `create` is `false` and the session does not exist, `SessionNotFoundException` is thrown. By default, a local, unsecure session is created.

## Usage

To create an unsecure, distributed session object, for example, use the following statement in your JServlet code:

```
oracle.OAS.servlet.http.HttpSession sess =  
    oracle.OAS.servlet.ServletRequest.getSession(true, true, false)
```

## Parameters

`create` - If an associated session cannot be found and you want to create a new session, `true`. `false` if you do not want to create a session object if one cannot be found.

`distributed` - `true` if you want the created session to be distributed, `false` otherwise. This parameter is meaningless when `create` is `false`.

`secure` - `true` if you want the created session to be secure, `false` otherwise. This parameter is meaningless when `create` is `false`.

## Returns

The session associated with this request or null if `create` was `false` and no valid session is associated with this request.

---

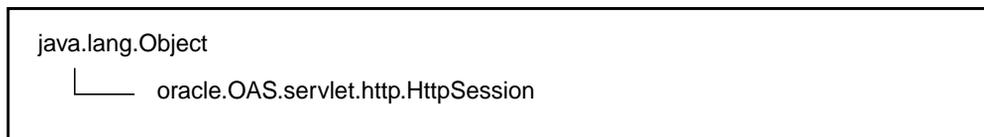
## HttpSession class

```
public class HttpSession  
extends Object
```

### Description

This class provides the user API for OAS Jweb session state management service. See Chapter 4, “Using the JWeb Toolkit” in *Developer’s Guide: JWeb Applications* for more information about using sessions.

**Figure 2–1 Inheritance for the HttpSession class**



## getCreationTime

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public long getCreationTime() throws Exception
```

### Purpose

Returns the time at which this session was created in milliseconds since January 1, 1970, 00:00:00GMT.

### Returns

The session creation time.

## getId

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public String getId()
```

### Purpose

Returns the identifier assigned to this session. Session identifier is a global unique string that is created and maintained by the session management service.

### Returns

The session identifier.

## getLastAccessedTime

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public long getLastAccessedTime() throws Exception
```

### Purpose

Returns the last time the client sent a request carrying the identifier assigned to the session or -1 if the session is new. Time is expressed as milliseconds since 00:00:00GMT January 1, 1970.

### Returns

The last time the client accessed the session.

## getMaxInactiveInterval

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public int getMaxInactiveInterval() throws Exception
```

### Purpose

Returns the maximum amount of time, in seconds, that a session is guaranteed to be maintained in the cache without a request from the client. After the maximum inactive time, the session will be removed from the system and future attempts to access the session will result in `SessionNotFoundException`s.

### Returns

The maximum inactive interval for this session.

## getValue

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public Object getValue(String name) throws Exception
```

### Purpose

Returns the object bound to a given name in the session. Returns null if there is no such binding.

### Parameters

name - the state object name

**Returns**

The state object.

**getValueNames****Class**

oracle.OAS.servlet.http.HttpSession

**Syntax**

```
public String[] getValueNames() throws Exception
```

**Purpose**

Returns an array of the names of all the values bound into this session.

**Returns**

An array of the names of all the values bound into this session.

**invalidate****Class**

oracle.OAS.servlet.http.HttpSession

**Syntax**

```
public void invalidate() throws Exception
```

**Purpose**

Destroys the session identified by this instance including all state objects associated with the session.

## isNew

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public boolean isNew() throws Exception
```

### Purpose

isNew returns a boolean value indicating whether this session is considered to be new. A session is considered to be new if it has been created by the server and not received from the client as part of this request.

### Returns

true if this session is new, false otherwise.

## putValue

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public void putValue(  
    String      name,  
    Object      obj) throws Exception
```

### Purpose

Binds the specified object into the session with the given name. Name is an application defined string used to identify the state object. It is used by subsequent getValue and removeValue calls.

### Parameters

name - the session state object name

obj - the session state object, the session state object can't be null

## removeValue

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public void removeValue(String name) throws Exception
```

### Purpose

Remove the named state object from the session. If there is no object bound to the given name, this method does nothing.

### Parameters

name - the state object name

## setMaxInactiveInterval

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public int setMaxInactiveInterval(int interval) throws Exception
```

### Purpose

Sets the amount of time that a session can be inactive before session is allowed to expire it.

### Parameters

the - inactive interval in seconds

### Returns

The inactive interval in seconds

## removeValue

### Class

oracle.OAS.servlet.http.HttpSession

### Syntax

```
public void removeValue(String name) throws Exception
```

### Purpose

Remove the named state object from the session. If there is no object bound to the given name, this method does nothing.

### Parameters

name - the state object name

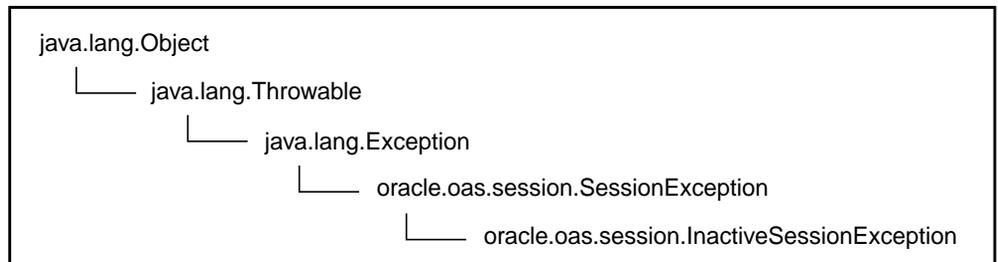
---

## InactiveSessionException class

public class **InactiveSessionException**  
extends SessionException

This exception is raised when the session referenced by the session id supplied is not active. This indicates that a stale reference to a Session object was used. This can be corrected by using the getSession method to get a new session object.

*Figure 2–2 Inheritance for the InactiveSessionException class*



## InactiveSessionException constructor

### Class

oracle.oas.session.InactiveSessionException

### Syntax

```
public InactiveSessionException()
```

```
public InactiveSessionException(String s)
```

### Purpose

Creates a new InactiveSessionException with the specified detailed message.

### Parameters

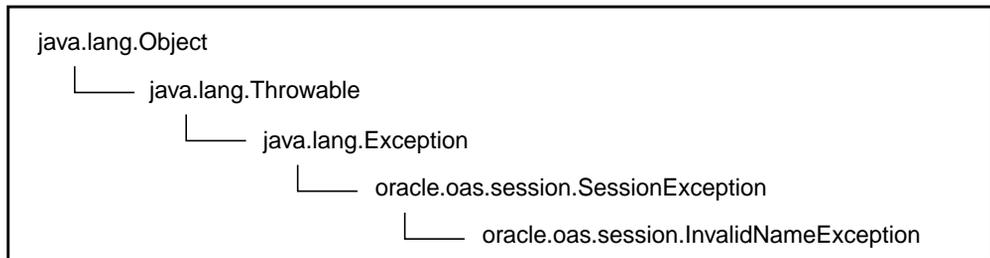
s - the detail message

## InvalidNameException class

**public class InvalidNameException**  
extends SessionException

This exception is raised when the session state object has either a null name or is reserved by the session state service.

*Figure 2–3 Inheritance for the InvalidNameException class*



### Variable

Name	Syntax	Description
base	public Exception base	

## InvalidNameException constructor

### Class

oracle.oas.session.InvalidNameException

### Syntax

```
public InvalidNameException()
```

```
public InvalidNameException(String s)
```

### **Purpose**

Creates a new InvalidNameException with the specified detailed message.

### **Parameters**

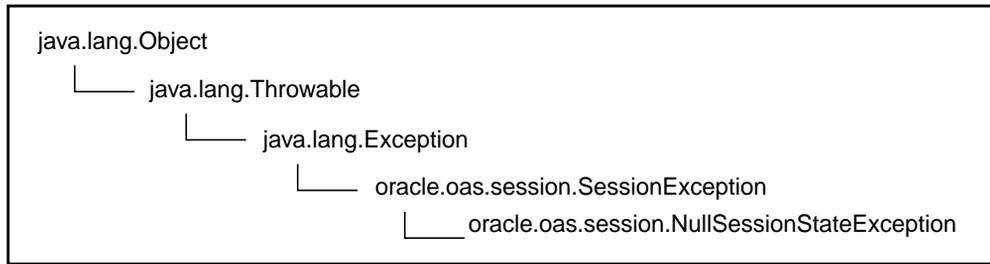
s - the detail message

## NullSessionStateException class

**public class NullSessionStateException**  
extends SessionException

This exception is raised when an attempt to store a null object is made. The session state management system does not allow the storage of null objects. This exception typically indicates a problem in the application.

*Figure 2–4 Inheritance for the NullSessionStateException class*



## NullSessionStateException constructor

### Class

`oracle.oas.session.NullSessionStateException`

### Syntax

```
public NullSessionStateException()
```

```
public NullSessionStateException(String s)
```

### Purpose

Creates a new `NullSessionStateException` with the specified detailed message.

### Parameters

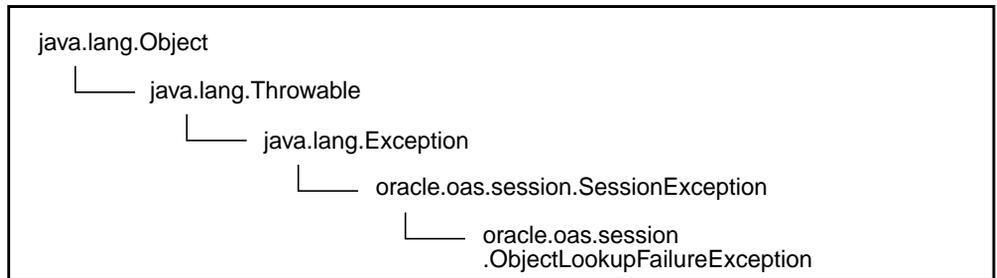
`s` - the detail message

## ObjectLookupFailureException class

```
public class ObjectLookupFailureException
extends SessionException
```

This exception is raised when an internal problem occurs that prevents the lookup of an object associated with the session. This is not a recoverable error. The session should be terminated and an exception returned to the user. Oracle Support should be notified if this exception is raised.

*Figure 2-5 Inheritance for the ObjectLookupFailureException class*



### Variable

Name	Syntax	Description
<code>base</code>	<code>public Exception base</code>	

## ObjectLookupFailureException constructor

### Class

```
oracle.oas.session.ObjectLookupFailureException
```

### Syntax

```
public ObjectLookupFailureException()
```

```
public ObjectLookupFailureException(String s)
```

### **Purpose**

Creates a new ObjectLookupFailureException with the specified detailed message.

### **Parameters**

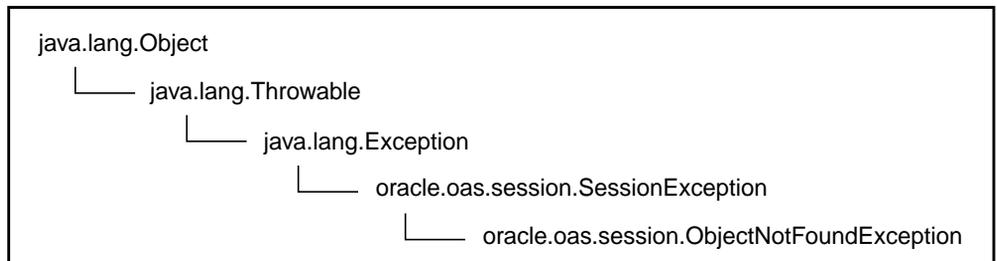
s - the detail message

## ObjectNotFoundException class

public class **ObjectNotFoundException**  
 extends SessionException

This exception is raised when the requested object is not a part of the session. This can occur when either the object was never stored in the session or the argument name is incorrect. In most cases, the application should handle this exception.

*Figure 2–6 Inheritance for the ObjectNotFoundException class*



### Variable

Name	Syntax	Description
<b>base</b>	<code>public Exception base</code>	

## ObjectNotFoundException constructor

### Class

`oracle.oas.session.ObjectNotFoundException`

### Syntax

`public ObjectNotFoundException()`

`public ObjectNotFoundException(String s)`

### **Purpose**

Creates a new `ObjectNotFoundException` with the specified detailed message.

### **Parameters**

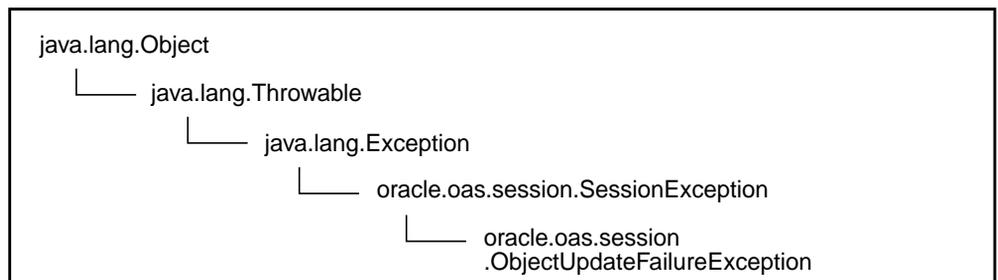
s - the detail message

## ObjectUpdateFailureException class

```
public class ObjectUpdateFailureException
extends SessionException
```

This exception is raised when the backup node fails to update the object.

*Figure 2–7 Inheritance for the ObjectUpdateFailureException class*



### Variable

Name	Syntax	Description
<code>base</code>	<code>public Exception base</code>	

## ObjectUpdateFailureException constructor

### Class

`oracle.oas.session.Exception`

### Syntax

```
public ObjectUpdateFailureException()
```

```
public ObjectUpdateFailureException(String s)
```

### **Purpose**

Creates a new ObjectUpdateFailureException with the specified detailed message.

### **Parameters**

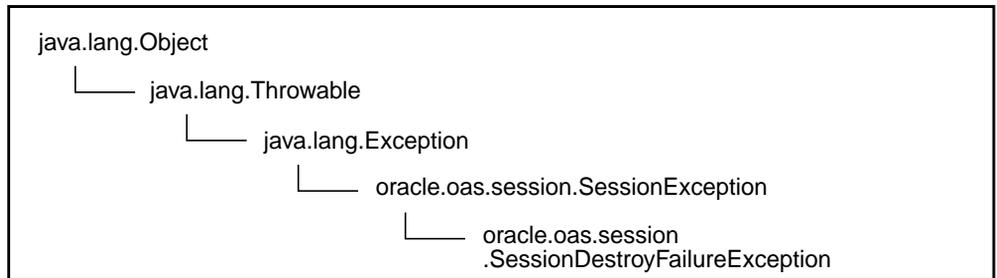
s - the detail message

## SessionDestroyFailureException class

public class **SessionDestroyFailureException**  
 extends SessionException

This exception is raised when an internal errors occurs while trying to invalidate a session. This exception, in isolation, is not a serious problem since the session is being destroyed, but may indicate a more serious problem elsewhere in the system. Oracle Support should be notified if this exception is raised.

**Figure 2–8 Inheritance for the SessionDestroyFailureException class**



### Variable

Name	Syntax	Description
<code>base</code>	<code>public Exception base</code>	

## SessionDestroyFailureException constructor

### Class

`oracle.oas.session.SessionDestroyFailureException`

### Syntax

```
public SessionDestroyFailureException()
```

```
public SessionDestroyFailureException(String s)
```

### **Purpose**

Creates a new `SessionDestroyFailureException` with the specified detailed message.

### **Parameters**

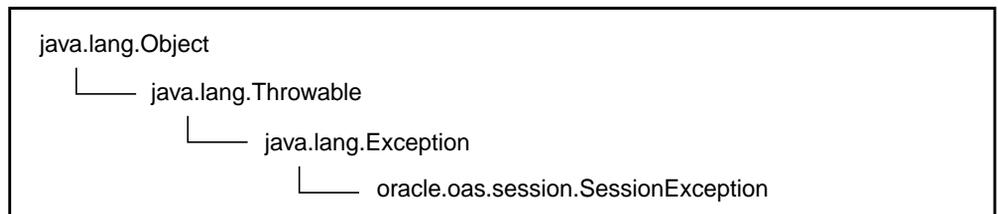
s - the detail message

## SessionException class

**public class SessionException**  
extends Exception

This is a generic exception for the session management system. All other session exceptions are extensions of this class.

**Figure 2–9 Inheritance for the SessionException class**



### Variable

Name	Syntax	Description
<b>base</b>	<code>public Exception base</code>	

## SessionException constructor

### Class

`oracle.oas.session.SessionException`

### Syntax

```
public SessionException()
```

```
public SessionException(String s)
```

```
public SessionException(
    String      s,
    Exception   ex)
```

### **Purpose**

Creates a new SessionException with the specified detailed message.

### **Parameters**

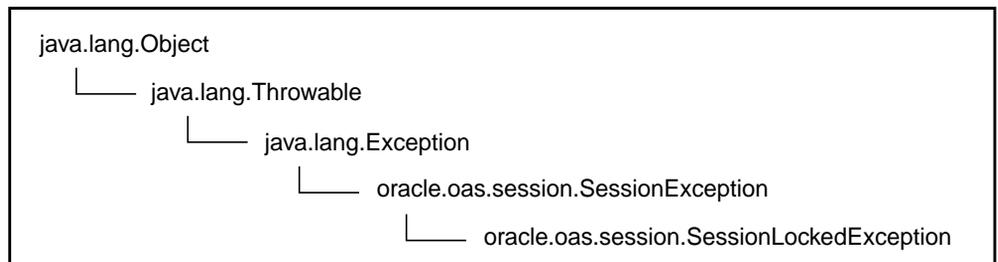
s - the detail message

## SessionLockedException class

public class **SessionLockedException**  
 extends SessionException

This exception indicates an internal problem with session locking. Oracle Support should be notified if this exception is raised.

**Figure 2–10 Inheritance for the SessionLockedException class**



### Variable

Name	Syntax	Description
<b>base</b>	public Exception base	

## SessionLockedException constructor

### Class

oracle.oas.session.SessionLockedException

### Syntax

```
public SessionLockedException()
```

```
public SessionLockedException(String s)
```

### **Purpose**

Creates a new `SessionLockedException` with the specified detailed message.

### **Parameters**

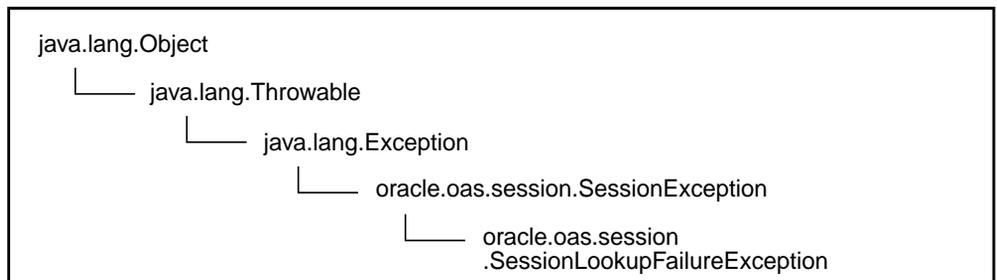
s - the detail message

## SessionLookupFailureException class

public class **SessionLookupFailureException**  
extends SessionException

This exception is raised when an internal error occurs while trying to locate a session. Oracle Support should be notified if this exception is raised.

**Figure 2–11 Inheritance for the SessionLookupFailureException class**



### Variable

Name	Syntax	Description
<b>base</b>	<code>public Exception base</code>	

## SessionLookupFailureException constructor

### Class

`oracle.oas.session.SessionLookupFailureException`

### Syntax

```
public SessionLookupFailureException()
```

```
public SessionLookupFailureException(String s)
```

### **Purpose**

Creates a new `SessionLookupFailureException` with the specified detailed message.

### **Parameters**

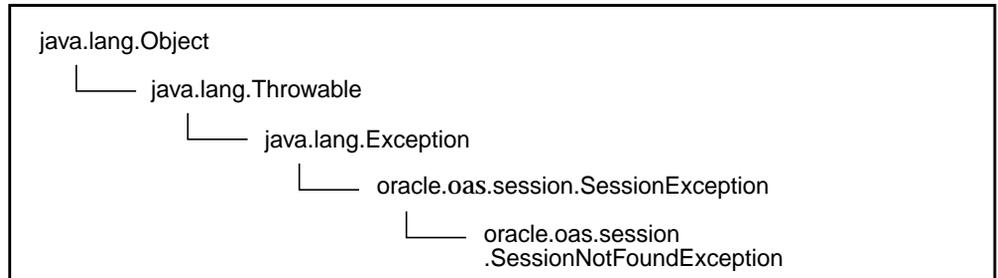
s - the detail message

## SessionNotFoundException class

public class **SessionNotFoundException**  
extends `SessionException`

This exception is raised when the requested session cannot be located. Either the session may have timed out or the session id may not be valid. The user should be notified that a new session needs to be started.

**Figure 2–12 Inheritance for the SessionNotFoundException class**



### Variable

Name	Syntax	Description
<code>base</code>	<code>public Exception base</code>	

## SessionNotFoundException constructor

### Class

`oracle.oas.session.SessionNotFoundException`

### Syntax

```
public SessionNotFoundException()
```

```
public SessionNotFoundException(String s)
```

### **Purpose**

Creates a new `SessionNotFoundException` with the specified detailed message.

### **Parameters**

s - the detail message

---

## oracle.owas.wrb.WRB package

This chapter contains reference pages for the following classes:

- [WRBRunnable class](#)
  - [WRBRunnable constructor](#)
  - [getWRBContext](#)
  - [run](#)
  - [setWRBContext](#)

This chapter contains reference pages for the following exception classes:

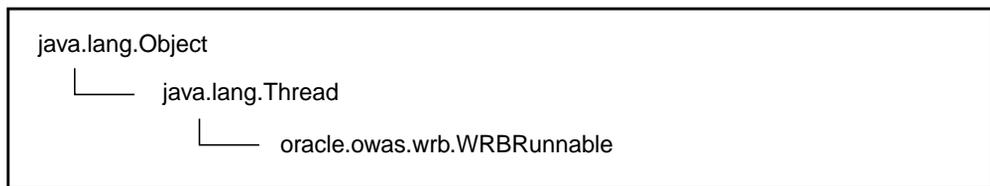
- [WRBNotRunningException class](#)
  - [WRBNotRunningException constructor](#)
- [WRBRuntimeException class](#)
  - [WRBRuntimeException constructor](#)

## WRBRunnable class

public class **WRBRunnable**  
extends Thread

This class essentially provides a way for adopting user threads. It is used by multi-threaded java applications.

*Figure 3–1 Inheritance for the WRBRunnable class*



## WRBRunnable constructor

### Class

oracle.owas.wrb.WRBRunnable

### Syntax

```
public WRBRunnable(Runnable userRunnable) throws WRBNotRunningException
```

### Purpose

Creates an instance of WRBRunnable. A WRBNotRunningException will be thrown if it failed to obtain a valid WRB Context. For example, this can be a result of running this class outside of OAS platform.

### Parameters

userRunnable - user defined Runnable

## getWRBContext

### Class

oracle.owas.wrb.WRBRunnable

### Syntax

```
public synchronized long getWRBContext()
```

### Returns

The WRB context.

## run

### Class

oracle.owas.wrb.WRBRunnable

### Syntax

```
public void run()
```

### Purpose

Runs the user thread. A runtime exception will be thrown if an invalid wrb context is encountered.

### Overrides

run in class Thread.

## setWRBContext

### Class

oracle.owas.wrb.WRBRunnable

### Syntax

```
public synchronized void setWRBContext(long wrbContext) throws  
WRBNotRunningException
```

### Purpose

Sets the WRB context to be associated with this Runnable. The WRB context must be retrieved from a main thread.

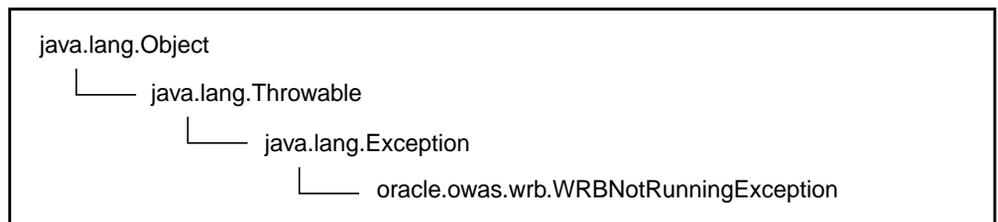
---

## WRBNotRunningException class

```
public class WRBNotRunningException
extends Exception
```

This class represents an exception when WRB is not running.

**Figure 3–2** Inheritance for the *WRBNotRunningException* class



## WRBNotRunningException constructor

### Class

`oracle.owas.wrb.WRBNotRunningException`

### Syntax

```
public WRBNotRunningException(String s)
```

### Purpose

Creates a new `WRBNotRunningException` with the specified detail message.

### Parameters

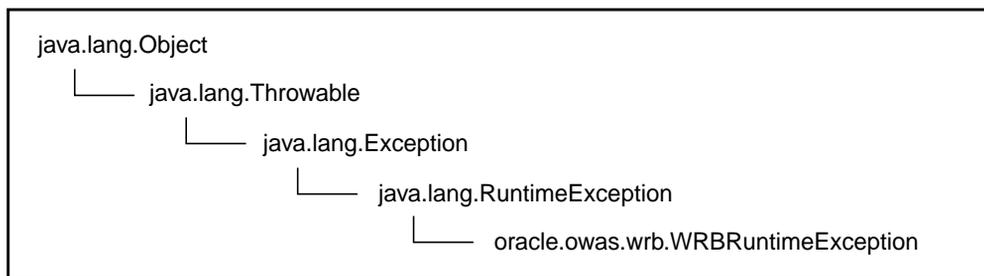
`s` - the detail message

## WRBRuntimeException class

public class **WRBRuntimeException**  
extends RuntimeException

This class represents an exception has occurred in WRB.

*Figure 3–3 Inheritance for the WRBRuntimeException class*



## WRBRuntimeException constructor

### Class

`oracle.owas.wrb.WRBRuntimeException`

### Syntax

```
public WRBRuntimeException(String s)
```

### Purpose

Creates a new `WRBRuntimeException` with the specified detail message.

### Parameters

`s` - the detail message

---

## oracle.OAS.Services.ICX package

This chapter contains reference pages for the following classes:

- [ICXRequest class](#)
  - [ICXRequest constructor](#)
  - [connect](#)
  - [enableTransaction, disableTransaction](#)
  - [setAuthInfo](#)
  - [setContent](#)
  - [setContents](#)
  - [setHeader](#)
  - [setHeaders](#)
  - [setMethod](#)
  - [setWalletInfo](#)
- [ICXResponse class](#)
  - [ICXResponse constructor](#)
  - [finalize](#)
  - [getContent](#)
  - [getHeader](#)
  - [getHeaders](#)
  - [getHTTPVersion](#)

- 
- `getRealm`
  - `getReasonPhrase`
  - `getStatusCode`
  - `usingProxy`
  - `ICXInitFailedException` class
    - `ICXInitFailedException` constructor
  - `IncompatibleWithProtocolException` class
    - `IncompatibleWithProtocolException` constructor

---

## ICXRequest class

```
public class ICXRequest
extends Object
```

This class is the ICX request class. It contains methods for setting up the request and for executing the request.

**Figure 4–1 Inheritance for the ICXRequest class**

```
java.lang.Object
└── oracle.OAS.Services.ICX.ICXRequest
```

### Example

The sequence of calls that you would invoke to send an ICX request is as follows:

```
// import packages
import oracle.owas.wrb.services.http.*;
import oracle.OAS.Services.ICX.*;
import java.io.*;

// set up the ICXRequest object
ICXRequest icxreq = new ICXRequest("http://machine/path");

// set up properties for the request (for example, method, headers, content,
// security, transaction)
// send the request and get an ICXResponse object
ICXResponse response = icxreq.connect();
if (icxresp == null) {
    // if the response is null, then return an error message and return.
    return;
}
```

## ICXRequest constructor

### Class

oracle.OAS.Services.ICX.ICXRequest

### Syntax

```
public ICXRequest(String url) throws MalformedURLException,  
ICXInitFailedException
```

```
public ICXRequest(URL url) throws MalformedURLException,  
ICXInitFailedException
```

### Purpose

Constructs a ICX Request Object from a url string or a url object.

### Example

```
// set up an ICX request  
try {  
    icxreq = new ICXRequest(url);  
} catch (java.net.MalformedURLException e) {  
    System.err.println("Invalid URL");  
    // send appropriate error message to client browser; something like  
    // "Unable to ccomplete your request at this time. Please  
    // try again."  
} catch (ICXInitFailedException e) {  
    System.err.println("Unable to create an ICX request");  
}
```

## connect

### Class

oracle.OAS.Services.ICX.ICXRequest

**Syntax**

```
public ICXResponse connect()
```

**Purpose**

Connects to the URL.

**Returns**

An ICXResponse object.

**Example**

```
// send the ICX request
ICXResponse icxresp = icxreq.connect();
if (icxresp==null) {
    // if the response is null, then return an error message and return.
    return;
}
```

**enableTransaction, disableTransaction****Class**

oracle.OAS.Services.ICX.ICXRequest

**Syntax**

```
public void enableTransaction()
```

```
public void disableTransaction()
```

**Purpose**

Enable or disable a transaction for this request.

## setAuthInfo

### Class

oracle.OAS.Services.ICX.ICXRequest

### Syntax

```
public void setAuthInfo(  
    String      name,  
    String      password)
```

### Purpose

Sets the username and password for the request.

### Example

```
// pull out the username and password from the headers.  
// Retrieve "user" and "password" from the HTTP header.  
user = curRequest.getHeader("user");  
password = curRequest.getHeader("password");  
  
// set the username and password.  
icxreq.setAuthInfo(user, password);
```

## setContent

### Class

oracle.OAS.Services.ICX.ICXRequest

### Syntax

```
public void setContent(  
    String      name,  
    String      value)
```

**Purpose**

Set the request content.

**setContents****Class**

oracle.OAS.Services.ICX.ICXRequest

**Syntax**

```
public void setContents(Hashtable contents) throws  
IllegalArgumentException
```

**Purpose**

Append contents presented by the hashtable to the existing set of contents.

**setHeader****Class**

oracle.OAS.Services.ICX.ICXRequest

**Syntax**

```
public void setHeader(  
    String      name,  
    String      value)
```

**Purpose**

Set the request header.

## setHeaders

### Class

oracle.OAS.Services.ICX.ICXRequest

### Syntax

```
public void setHeaders(Hashtable headers) throws  
IllegalArgumentException
```

### Purpose

Append headers presented by the hashtable to the existing set of headers.

## setMethod

### Class

oracle.OAS.Services.ICX.ICXRequest

### Syntax

```
public void setMethod(String method) throws ProtocolException
```

### Purpose

Set the request method.

## setWalletInfo

### Class

oracle.OAS.Services.ICX.ICXRequest

**Syntax**

```
public void setWalletInfo(  
    String      walletloc,  
    String      password) throws IncompatibleWithProtocolException
```

**Purpose**

Set the wallet location and password for the request.

**Throws**

[IncompatibleWithProtocolException](#)

---

## ICXResponse class

```
public class ICXResponse  
extends Object
```

This class is the ICX response class. It contains methods for retrieving various parts of the response.

*Figure 4–2 Inheritance for the ICXResponse class*



## ICXResponse constructor

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
public ICXResponse()
```

## finalize

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
protected void finalize() throws Throwable
```

**Overrides**

finalize in class Object

**getContent****Class**

oracle.OAS.Services.ICX.ICXResponse

**Syntax**

```
public InputStream getContent()
```

**Returns**

An InputStream of content.

**getHeader****Class**

oracle.OAS.Services.ICX.ICXResponse

**Syntax**

```
public Object getHeader(String name) throws NoSuchElementException
```

**Returns**

The header value associated with the name.

## getHeaders

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
public Hashtable getHeaders()
```

### Returns

A hashtable representing all the headers.

## getHTTPVersion

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
public String getHTTPVersion()
```

### Returns

The HTTP version.

## getRealm

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
public String getRealm()
```

**Returns**

Returns the realm.

**getReasonPhrase****Class**

oracle.OAS.Services.ICX.ICXResponse

**Syntax**

```
public String getReasonPhrase()
```

**Returns**

The reason phrase.

**getStatusCode****Class**

oracle.OAS.Services.ICX.ICXResponse

**Syntax**

```
public int getStatusCode()
```

**Returns**

Returns the status code.

## usingProxy

### Class

oracle.OAS.Services.ICX.ICXResponse

### Syntax

```
public boolean usingProxy()
```

### Returns

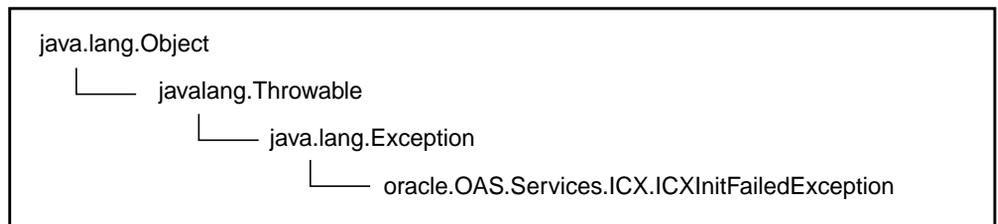
true if proxy is used for the request. false otherwise.

---

## ICXInitFailedException class

```
public class ICXInitFailedException  
extends Exception
```

**Figure 4–3** Inheritance for the *ICXInitFailedException* class



## ICXInitFailedException constructor

### Class

`oracle.OAS.Services.ICX.ICXInitFailedException`

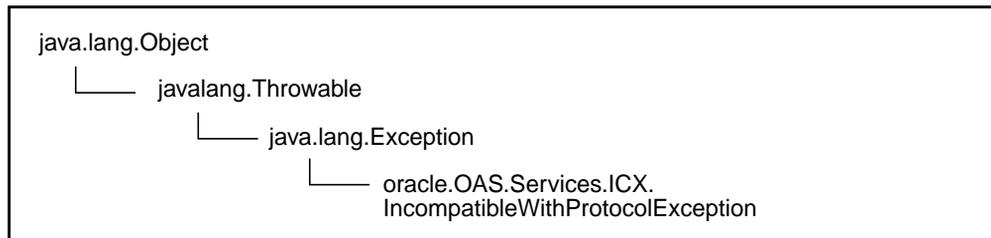
### Syntax

```
public ICXInitFailedException()
```

## IncompatibleWithProtocolException class

```
public class IncompatibleWithProtocolException  
extends Exception
```

*Figure 4–4 Inheritance for the IncompatibleWithProtocolException class*



## IncompatibleWithProtocolException constructor

### Class

oracle.OAS.Services.ICX.IncompatibleWithProtocolException

### Syntax

```
public IncompatibleWithProtocolException()
```

---

---

## oracle.lang package

This chapter contains reference pages for the following classes:

- [NLS class](#)
  - [setCurrentCharSet, currentCharSet](#)
  - [length](#)
  - [NLS2Unicode, Unicode2NLS](#)
  - [NLSBufferSize](#)

This chapter contains reference pages for the following exception classes:

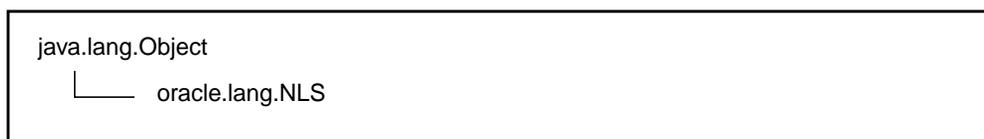
- [NLSRuntimeException class](#)
  - [NLSRuntimeException constructor](#)
- [CharacterSetException class](#)
  - [CharacterSetException constructor](#)

## NLS class

**public abstract class NLS**  
extends Object

This class is obsolete. It is preserved to provide backward compatibility. Users should not use it any more.

*Figure 5–1 Inheritance for the NLS class*



### Constants

Name	Syntax	Description
ASCII	<code>public static final int ASCII</code>	Constant represents 7-bit ASCII character set.

## setCurrentCharSet, currentCharSet

### Class

oracle.lang.NLS

### Syntax

```
public static synchronized void setCurrentCharSet(int charSet)

public static int currentCharSet()
```

### Purpose

Sets or returns the current character set. The default character set is 7-bit ASCII.

**Parameters**

charSet - the character set to set to

**length****Class**

oracle.lang.NLS

**Syntax**

```
public static native int length(  
    byte          str[],  
    int           length,  
    int           charSet)
```

**Purpose**

Returns the length (character count) of a NLS string.

**Parameters**

str - the NLS string

length - the length of this NLS String (number of bytes)

charSet - the character set of the NLS string

**Returns**

The length of the NLS string.

**NLS2Unicode, Unicode2NLS****Class**

oracle.lang.NLS

## Syntax

```
public static native int NLS2Unicode(  
    byte        src[],  
    char        dst[],  
    int         srcBegin,  
    int         srcEnd,  
    int         dstBegin,  
    int         charSet)
```

```
public static native int Unicode2NLS(  
    char        src[],  
    byte        dst[],  
    int         srcBegin,  
    int         srcEnd,  
    int         dstBegin,  
    int         charSet)
```

## Purpose

Converts a NLS string to its Unicode equivalent or a Unicode string to its NLS equivalent, respectively.

## Parameters

src - the source string

dst - the target string

srcBegin - begin copy at this offset in the source string

srcEnd - stop copy at this offset in the source string (exclusive)

dstBegin - offset into the target string

charSet - the character set of the source string

## Returns

The length of NLS string (number of bytes).

## NLSBufferSize

### Class

oracle.lang.NLS

### Syntax

```
public static native int NLSBufferSize(  
    char          str[],  
    int          charSet)
```

### Purpose

Determines the minimum number of bytes needed to store the NLS-equivalent of a Unicode string.

### Parameters

str - the source Unicode string

charSet - the target character set

### Returns

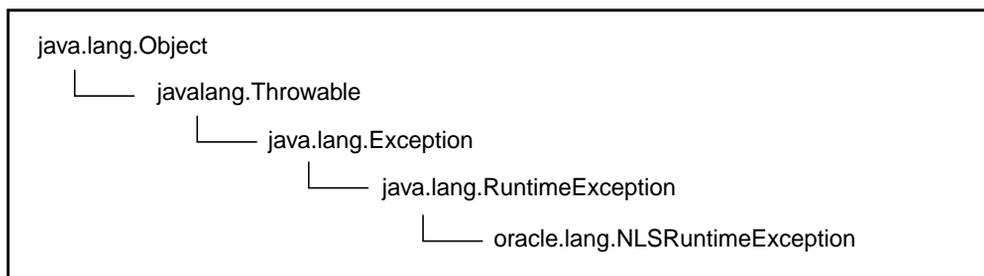
The number of bytes needed.

## NLSRuntimeException class

```
public class NLSRuntimeException  
extends RuntimeException
```

This class represents an exception has occurred in Oracle NLS runtime library.

**Figure 5–2 Inheritance for the NLSRuntimeException class**



## NLSRuntimeException constructor

### Class

oracle.lang.NLSRuntimeException

### Syntax

```
public NLSRuntimeException(String s)
```

### Purpose

Creates a new NLSRuntimeException with the specified detail message.

### Parameters

s - the detail message

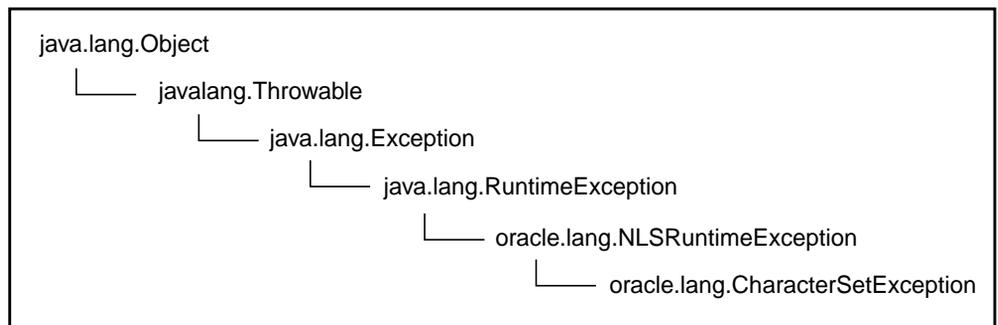
---

## CharacterSetException class

```
public class CharacterSetException
extends NLSRuntimeException
```

This class represents an exception has occurred in Oracle NLS runtime library due to problem in character set.

**Figure 5–3 Inheritance for the CharacterSetException class**



## CharacterSetException constructor

### Class

`oracle.lang.CharacterSetException`

### Syntax

```
public CharacterSetException(String s)
```

### Purpose

Creates a new `CharacterSetException` with the specified detail message.

### Parameters

`s` - the detail message



---

---

## oracle.owas.nls package

This chapter contains reference pages for the following classes:

- [CharacterSet class](#)
  - [CharacterSet constructor](#)
  - [bytesToBytes](#)
  - [bytesToChars](#)
  - [bytesToString](#)
  - [charsToBytes](#)
  - [equals](#)
  - [getMaxBytesPerChar](#)
  - [getName](#)
  - [minByteSize \(Java character strings\)](#)
  - [minByteSize \(other character sets\)](#)
  - [minCharSize](#)
  - [stringToBytes](#)
- [CharacterSetManager class](#)
  - [getCharacterSet](#)
- [NLS\\_LANG class](#)
  - [NLS\\_LANG constructor](#)
  - [getCharacterSet](#)

- 
- [getContentLanguage](#)
  - [getCurrent](#)
  - [getIANACharacterSet](#)
  - [getISOLanguage](#)
  - [getISOTerritory](#)
  - [getLanguage](#)
  - [getOracleCharacterSet](#)
  - [getOracleLanguage](#)
  - [getOracleTerritory](#)
  - [getTerritory](#)
  - [getValue](#)
  - [setAcceptLanguage](#)
  - [setCharacterSet](#)
  - [setLanguage](#)
  - [setTerritory](#)
  - [setValue](#)
  - [toString](#)

This chapter contains reference pages for the following exception classes:

- [CharacterSetException class](#)
  - [CharacterSetException constructor](#)
- [NLSRuntimeException class](#)
  - [NLSRuntimeException constructor](#)

---

## CharacterSet class

public abstract class **CharacterSet**  
extends Object

This class defines the interface of a character set which is used to handling character set encoding and conversion.

**Figure 6–1 Inheritance for the CharacterSet class**



This class defines the interface of a character set which is used to handling character set encoding and conversion.

Users who wish to implement a character set must implement the following methods:

- String getName()
- int getMaxBytesPerChar()
- int minCharSize(int byteCount)
- int charsToBytes(char src[], byte dst[], int srcBegin, int length, int dstBegin)
- int bytesToChars(byte src[], char dst[], int srcBegin, int length, int dstBegin)

Users can obtain a character set implemented by Oracle's NLS package from CharacterSetManager as follows:

```
// Gets an Oracle CharacterSet
CharacterSet charset = CharacterSetManager.getCharacterSet("US7ASCII");
```

Or users can also implement their own character sets as follows:

```
// Creates my own CharacterSet
CharacterSet charset = new MyCharacterSet();
```

And use the character set to convert a Java String (in Unicode) to native character set encoding format:

```
String hello = "Hello";  
byte  nativeHello[] = charset.stringToBytes(hello);
```

## CharacterSet constructor

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public CharacterSet()
```

### Purpose

Creates a new CharacterSet.

## bytesToBytes

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public int bytesToBytes(  
    byte      src[],  
    byte      dst[],  
    int       srcBegin,  
    int       length,  
    CharacterSet srcCharSet,  
    int       dstBegin)
```

## Purpose

Converts a string of this CharacterSet to another CharacterSet. This method calls the source character set to convert the source string to Java chars and then the target character set to convert it to the target character set. Subclass may want to override this method to convert the string more efficiently.

## Parameters

src - the source byte string

dst - the target byte string buffer

srcBegin - begin position of src

length - the length of the source string from begin position

srcCharSet - the character set of the source string

dstBegin - begin position of dst

## Returns

Length of target NLS string (number of bytes).

## bytesToChars

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public abstract int bytesToChars(  
    byte          src[],  
    char         dst[],  
    int          srcBegin,  
    int          length,  
    int          dstBegin)
```

### Purpose

Converts a string in this CharacterSet to a Java char string.

### Parameters

src - the source byte string

dst - the target char string buffer

srcBegin - begin position of src

length - the length of the source string from begin position

dstBegin - begin position of dst

### Returns

The number of chars written to dst.

## bytesToString

### Class

oracle.ows.nls.CharacterSet

### Syntax

```
public String bytesToString(byte bytes[])
```

```
public String bytesToString(  
    byte        bytes[],  
    int         begin,  
    int         length)
```

### Purpose

Converts a string from this CharacterSet to a Java string.

### Parameters

bytes - the source byte array

begin - the offset to begin

length - the length

**Returns**

The target Java string.

**charsToBytes****Class**

oracle.owas.nls.CharacterSet

**Syntax**

```
public abstract int charsToBytes(  
    char        src[],  
    byte        dst[],  
    int         srcBegin,  
    int         length,  
    int         dstBegin)
```

**Purpose**

Converts a Java char string to this CharacterSet.

**Parameters**

src - the source char string

dst - the target string buffer

srcBegin - begin position of src

length - the length of the source string from begin position

dstBegin - begin position of dst

**Returns**

The number of bytes written to dst.

## equals

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public boolean equals(CharacterSet charSet)
```

### Purpose

Returns the boolean result whether this character set is equivalent to another CharacterSet. This method sees if the character set is equal to this character set by comparing the IANA names.

### Parameters

charSet - the CharacterSet to compare with

### Returns

true if this character set is equivalent to another CharacterSet, false otherwise.

## getMaxBytesPerChar

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public abstract int getMaxBytesPerChar()
```

### Purpose

Returns the maximum number of bytes a character takes in this CharacterSet.

## getName

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public abstract String getName()
```

### Purpose

Returns the Internet Assigned Numbers Authority (IANA) name of this Character-Set.

## minByteSize (Java character strings)

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public int minByteSize(int charCount)
```

```
public int minByteSize(char chars[])
```

```
public int minByteSize(  
    char        chars[],  
    int         begin,  
    int         length)
```

### Purpose

Determines the number of bytes needed to store a Java character string in this character set object.

### Parameters

charCount - the number of characters in the Java string

chars - the Java string

begin - the beginning position of the string

length - the length of the string from begin position

### Returns

{ charcount | length() } \* [getMaxBytesPerChar\(\)](#)

---

---

**Note:** You may want to override this method in a subclass to provide a more accurate estimate.

---

---

## minByteSize (other character sets)

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public int minByteSize(  
    int          byteCount,  
    CharacterSet srcCharSet)
```

```
public int minByteSize(  
    byte        bytes[],  
    CharacterSet srcCharSet)
```

```
public int minByteSize(  
    byte        bytes[],  
    int         begin,  
    int         length,  
    CharacterSet srcCharSet)
```

## Purpose

Determines the minimum number of bytes needed to store a string in this character set.

## Parameters

bytes - the source string

begin - the begin position of the string

length - the length of the string from begin position

srcCharacterSet - the source character set

## Returns

The minimum number of bytes required to store a string in this character set.

---

---

**Note:** You may want to override this method in a subclass to provide a more accurate estimate.

---

---

## minCharSize

### Class

oracle.owas.nls.CharacterSet

### Syntax

```
public abstract int minCharSize(int byteCount)
```

```
public int minCharSize(byte bytes[])
```

```
public int minCharSize(  
    byte        bytes[],  
    int         begin,  
    int         length)
```

### **Purpose**

Determines the minimum number of chars needed to store a string in this CharacterSet in a Java string.

### **Parameters**

byteCount - the number of bytes of the source string

bytes - the source string

begin - the begin position of the string

length - the length of the string from begin position

### **Returns**

The number of characters needed.

## **stringToBytes**

### **Class**

oracle.owas.nls.CharacterSet

### **Syntax**

```
public byte[] stringToBytes(String string)
```

### **Purpose**

Converts a Java string to this CharacterSet.

### **Parameters**

string - the source Java string

### **Returns**

The target NLS string. The string (in byte[]) is NULL-terminated.

## CharacterSetManager class

```
public class CharacterSetManager
extends Object
```

This class manages Oracle's character sets. It is used to obtain Oracle character set.

**Figure 6–2 Inheritance for the CharacterSetManager class**



### Constants

Name	Syntax	Description
<b>IANA_CHARSET</b>	<code>public static final int IANA_CHARSET</code>	Internet Assigned Number Authority (IANA) character set.
<b>ORACLE_CHARSET</b>	<code>public static final int ORACLE_CHARSET</code>	Oracle character set

## getCharacterSet

### Class

oracle.owas.nls.CharacterSetManager

### Syntax

```
public static CharacterSet getCharacterSet()
```

```
public static CharacterSet getCharacterSet(String name)
```

```
public static CharacterSet getCharacterSet(
    int          type,
    String       name)
```

### **Parameters**

type - type of the character set: ORACLE\_CHARSET or IANA\_CHARSET

name - Oracle name of the character set if type is ORACLE\_CHARSET, IANA name of the character set if type is IANA\_CHARSET

### **Returns**

An Oracle character set used in the current NLS\_LANG environment.

## NLS\_LANG class

```
public class NLS_LANG
extends Object
```

This class encapsulates NLS\_LANG information. NLS\_LANG's format is `&lt;language>.&lt;territory>.&lt;character set>`. It also provides the mapping between Oracle's language, territory, character set and the ISO/IANA counter parts.

**Figure 6-3 Inheritance for the NLS\_LANG class**



### Constants

Name	Syntax	Description
<b>DEFAULT-NLS-<u>LANG</u></b>	<code>public static final String DEFAULT-NLS-<u>LANG</u></code>	Default NLS_LANG, which is AMERICAN_AMERICA.US7ASCII.

## NLS\_LANG constructor

### Class

```
oracle.owas.nls.NLS_LANG
```

### Syntax

```
public NLS_LANG()

public NLS_LANG(String NLS_LANG)

public NLS_LANG(
    String          acceptLanguage,
    String          NLS_LANG)
```

### Purpose

Constructs a new NLS\_LANG (with the NLS\_LANG value with the NLS\_LANG value, if specified).

---

---

**Note:** If *acceptLanguage* is passed as a parameter then a NLS\_LANG which is the best match of the HTTP's Accept-Language is constructed. If one cannot be found, NLS\_LANG is used instead.

---

---

### Parameters

*acceptLanguage* - the HTTP Accept-Language. If null, en (English) is assumed.

*NLS\_LANG* - the default NLS\_LANG to use if no match is found. If null, AMERICAN\_AMERICA.US7ASCII is assumed.

## getCharacterSet

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String getCharacterSet()
```

### Purpose

Returns NLS\_LANG character set.

## getContentLanguage

### Class

oracle.owas.nls.NLS\_LANG

**Syntax**

```
public String getContentLanguage()
```

**Purpose**

Returns the best-matched HTTP Content-Language of this NLS\_LANG.

**getCurrent****Class**

```
oracle.owas.nls.NLS_LANG
```

**Syntax**

```
public static NLS_LANG getCurrent()
```

**Purpose**

Returns a NLS\_LANG corresponds to the current NLS\_LANG setting in the environment.

**getIANACharacterSet****Class**

```
oracle.owas.nls.NLS_LANG
```

**Syntax**

```
public String getIANACharacterSet()
```

```
public static native String getIANACharacterSet(String charSet)
```

**Purpose**

Returns the IANA name of the character set.

### Parameters

charSet - Oracle name of a character set

## getISOLanguage

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String getISOLanguage()
```

```
public static native String getISOLanguage(String language)
```

### Purpose

Returns the ISO-3166 name of the language.

### Parameters

language - Oracle name of a national language

## getISOTerritory

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String getISOTerritory()
```

```
public static native String getISOTerritory(String territory)
```

### Purpose

Returns the ISO-639 name of an Oracle territory.

**Parameters**

territory - Oracle name of a territory

**getLanguage****Class**

oracle.owas.nls.NLS\_LANG

**Syntax**

```
public String getLanguage()
```

**Purpose**

Returns NLS\_LANG language.

**getOracleCharacterSet****Class**

oracle.owas.nls.NLS\_LANG

**Syntax**

```
public static native String getOracleCharacterSet(String name)
```

**Purpose**

Returns the Oracle name of an IANA character set.

**Parameters**

charSet - IANA name of a character set

## getOracleLanguage

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public static native String getOracleLanguage(String language)
```

### Purpose

Returns the Oracle name of an ISO-3166 national language.

### Parameters

language - ISO-3166 name of a national language

## getOracleTerritory

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public static native String getOracleTerritory(String territory)
```

### Purpose

Returns the Oracle name of an ISO-639 territory.

### Parameters

territory - ISO-639 name of a territory

## getTerritory

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String getTerritory()
```

### Purpose

Returns NLS\_LANG territory.

## getValue

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String getValue()
```

### Returns

NLS\_LANG.

## setAcceptLanguage

### Class

oracle.owas.nls.NLS\_LANG

## Syntax

```
public void setAcceptLanguage(  
    String      acceptLanguage,  
    String      NLS_LANG)
```

## Purpose

Sets the NLS\_LANG value which is the best match of the HTTP's Accept-Language. If one cannot be found, use the given NLS\_LANG instead. If NLS\_LANG is null, AMERICAN\_AMERICA.US7ASCII is assumed. If acceptLanguage is null, en (English) is assumed. In a multi threaded environment user should not attempt to use this method to change NLS\_LANG value in different threads.

## Parameters

acceptLanguage - the HTTP's Accept-Language

NLS\_LANG - the default NLS\_LANG to use if no match is found

## setCharacterSet

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public void setCharacterSet(String characterSet)
```

### Purpose

Sets NLS\_LANG character set.

## setLanguage

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public void setLanguage(String language)
```

### Purpose

Sets NLS\_LANG language.

## setTerritory

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public void setTerritory(String territory)
```

### Purpose

Sets NLS\_LANG territory.

## setValue

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public synchronized void setValue(String NLS_LANG)
```

### Purpose

Sets the NLS\_LANG value. If NLS\_LANG is null, AMERICAN\_AMERICA.US7ASCII is assumed.

### Parameters

NLS\_LANG - NLS\_LANG value

## toString

### Class

oracle.owas.nls.NLS\_LANG

### Syntax

```
public String toString()
```

### Purpose

Returns NLS\_LANG as a string in Oracle NLS\_LANG's format.

### Overrides

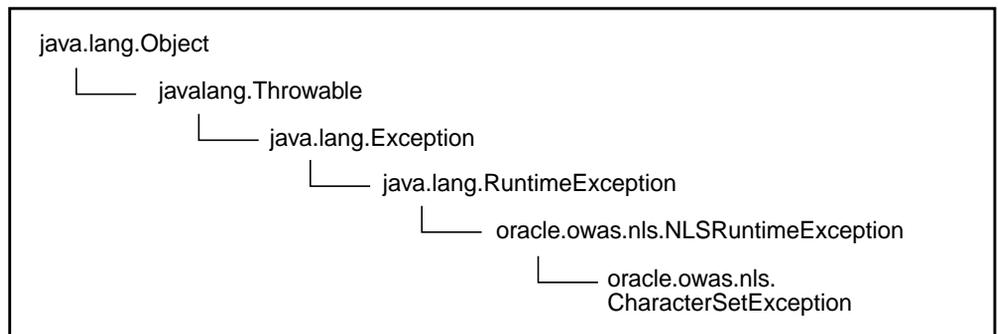
toString in class Object.

---

## CharacterSetException class

```
public class CharacterSetException
extends NLSRuntimeException
```

**Figure 6–4** Inheritance for the *CharacterSetException* class



## CharacterSetException constructor

### Class

`oracle.owas.nls.CharacterSetException`

### Syntax

```
public CharacterSetException(String s)
```

### Purpose

Creates a new `CharacterSetException` with the specified detail message.

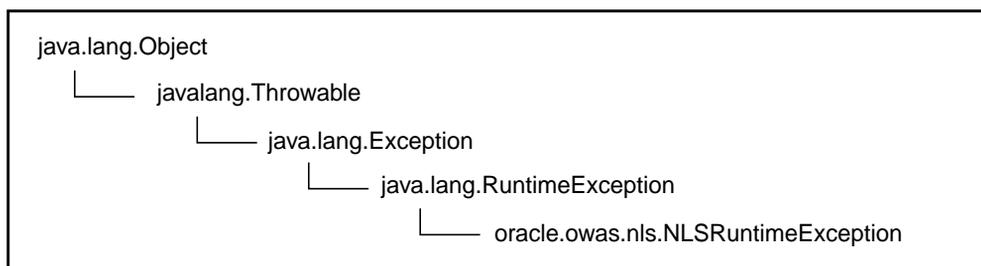
### Parameters

`s` - the detail message

## NLSRuntimeException class

```
public class NLSRuntimeException  
extends RuntimeException
```

*Figure 6–5 Inheritance for the NLSRuntimeException class*



## NLSRuntimeException constructor

### Class

oracle.owas.nls.NLSRuntimeException

### Syntax

```
public NLSRuntimeException(String s)
```

### Purpose

Creates a new NLSRuntimeException with the specified detail message.

### Parameters

s - the detail message

---

---

## oracle.plsql package

To pass data between Java and PL/SQL, we encapsulate the PL/SQL data types with Java wrapper classes. Primitive types in Java cannot represent PL/SQL types because PL/SQL types can have null values, which are not allowed in Java. So to represent PL/SQL types without losing the null value, they must be encapsulated.

These classes are derived from the `PValue` base class, which encapsulates the null attribute of PL/SQL values. Each individually derived class represents one or more related PL/SQL data types. For more information see Chapter 10, “pl2java” in *Developer’s Guide: JServlet Applications* for more information.

This chapter contains reference pages for the following classes:

- [PBoolean class](#)
  - [PBoolean constructor](#)
  - [booleanValue](#)
  - [copy](#)
  - [setValue](#)
  - [toString](#)
- [PByteArray class](#)
  - [PByteArray constructor](#)
  - [byteArrayValue](#)
  - [clone](#)
  - [copy](#)
  - [getByteArray](#)

- 
- length
  - setValue
  - toString
  - useBuffer
  - PDate class
    - PDate constructor
    - clone
    - copy
    - dateValue
    - getDate
    - setValue
    - toString
  - PDouble class
    - PDouble constructor
    - copy
    - doubleValue
    - intValue
    - setValue
    - toString
  - PInteger class
    - PInteger constructor
    - copy
    - intValue
    - setValue
    - toString
  - PStringBuffer class
    - PStringBuffer constructor
    - byteArrayValue

- 
- capacity
  - charArrayValue
  - clone
  - copy
  - getByteArray
  - getStringBuffer
  - getValue
  - length
  - setValue
  - stringBufferValue
  - stringValue
  - toCharArray
  - toString
  - useBuffer
  - PValue class
    - clone
    - copy
    - isNull
    - setNull
    - toString

This chapter contains reference pages for the following exception classes:

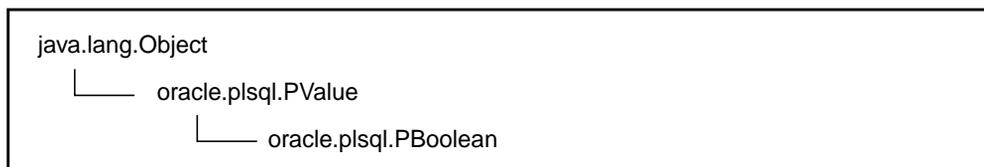
- NonUniformSizeException class
  - NonUniformSizeException constructor
- NullValueException class
  - NullValueException constructor
- PLSQLRuntimeException class
  - PLSQLRuntimeException constructor

## PBoolean class

```
public class PBoolean  
extends PValue
```

This class encapsulates PL/SQL booleans.

*Figure 7–1 Inheritance for the PBoolean class*



### See Also

- Boolean

## PBoolean constructor

### Class

oracle.plsql.PBoolean

### Syntax

```
public PBoolean()
```

```
public PBoolean(boolean value)
```

### Purpose

Constructs a PL/SQL boolean. If no parameters are specified then a null PL/SQL boolean will be constructed.

### Parameters

value - the initial value of the PBoolean

## booleanValue

### Class

oracle.plsql.PBoolean

### Syntax

```
public boolean booleanValue()
```

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the boolean value.

---

---

### Returns

The value of this PBoolean as a boolean.

### Throws

NullPointerException if the value is NULL.

### See Also

- [isNull](#) in [PValue class](#).

## copy

### Class

oracle.plsql.PBoolean

### Syntax

```
public void copy(PBoolean value)
```

### Purpose

Copies the PBoolean.

### Parameters

value - the PValue to copy from

## setValue

### Class

oracle.plsql.PBoolean

### Syntax

```
public void setValue(boolean value)
```

### Purpose

Sets the value of this PBoolean. To set this PBoolean to null, use [setNull\(\)](#).

### Parameters

value - the boolean value to set

### See Also

- [setNull](#) in *PValue* class.

## toString

### Class

oracle.plsql.PBoolean

### Syntax

```
public String toString()
```

**Returns**

The string representation of this PBoolean.

**Overrides**

[toString](#) in [PValue](#) class.

## PByteArray class

```
public class PByteArray  
extends PValue
```

This class encapsulates byte array for use in PL/SQL RAW and LONG RAW.

*Figure 7-2 Inheritance for the PByteArray class*



## PByteArray constructor

### Class

oracle.plsql.PByteArray

### Syntax

```
public PByteArray(int length)
```

```
public PByteArray(byte value[])
```

```
public PByteArray(  
    int          length,  
    byte         value[])
```

### Purpose

Constructs a PByteArray for RAW or LONG RAW. If only a byte parameter is passed then the given byte array will be used as a buffer.

### Parameters

length - the length of the byte array

---

value - the byte array to use or the initial content of the byte array

## byteArrayValue

### Class

oracle.plsql.PByteArray

### Syntax

```
public byte[] byteArrayValue()
```

### Purpose

Returns the value of this PByteArray as a byte array.

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the value.

---

---

### Returns

A copy of the byte array is returned.

### Throws

NullValueException if the value is NULL.

### See Also

- [isNull](#) in [PValue](#) class.

## clone

### Class

oracle.plsql.PByteArray

### Syntax

```
public Object clone()
```

### Purpose

Clone this PByteArray.

### Overrides

[clone](#) in [PValue](#) class.

## copy

### Class

oracle.plsql.PByteArray

### Syntax

```
public void copy(PByteArray value)
```

### Purpose

Copies the PByteArray.

### Parameters

value - the PValue to copy from

## getByteArray

### Class

oracle.plsql.PByteArray

## Syntax

```
public void getByteArray(byte buffer[])
```

## Purpose

Extracts the value of this PByteArray in a byte array.

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the value.

---

---

## Parameters

buffer - the byte array to store the value

## Throws

[NullValueException](#) if the value is NULL.

## See Also

- [isNull](#) in [PValue](#) class.

# length

## Class

oracle.plsql.PByteArray

## Syntax

```
public int length()
```

## Returns

The length of this PByteArray.

## setValue

### Class

oracle.plsql.PByteArray

### Syntax

```
public void setValue(byte value[])
```

### Purpose

Sets the value of this PByteArray. To set this PByteArray to null, either pass a null byte array or use [setNull\(\)](#).

### Parameters

value - the double value to set

### See Also

- [setNull](#) in [PValue](#) class.

## toString

### Class

oracle.plsql.PByteArray

### Syntax

```
public String toString()
```

### Returns

The string representation of this PByteArray.

### Overrides

[toString](#) in [PValue](#) class.

## useBuffer

### Class

oracle.plsql.PByteArray

### Syntax

```
public void useBuffer(boolean use)
```

### Purpose

Specifies whether this buffer should be used directly in PL/SQL execution. The default value is false. This may speed up the performance of PL/SQL execution by minimizing memory copy. However, if the same PL/SQL procedure or function is used repeatedly with different buffers, it will perform better by not using this buffer. In such a case, the wrapper class will allocate its own buffer to serve all calls.

### Parameters

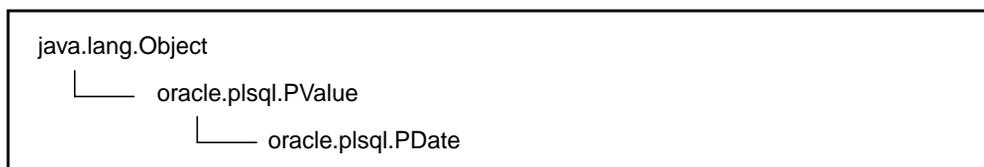
use - use this buffer or not

## PDate class

```
public class PDate  
extends PValue
```

This class encapsulates PL/SQL dates. It is implemented to look like `java.util.Date` as closely as possible.

**Figure 7–3 Inheritance for the PDate class**



## PDate constructor

### Class

`oracle.plsql.PDate`

### Syntax

```
public PDate()
```

```
public PDate(Date value)
```

### Purpose

Constructs a null PL/SQL date.

### Parameters

`value` - the initial value of the PDate object

## clone

### Class

oracle.plsql.PDate

### Syntax

```
public Object clone()
```

### Purpose

Clone this PDate.

### Overrides

[clone](#) in [PValue class](#).

## copy

### Class

oracle.plsql.PDate

### Syntax

```
public void copy(PDate value)
```

### Purpose

Copies the PDate.

### Parameters

value - the PValue to copy from

## dateValue

### Class

oracle.plsql.PDate

### Syntax

```
public Date dateValue()
```

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the Date value.

---

---

### Returns

The value of this PDate as a Date.

### Throws

NullPointerException if the value is NULL.

### See Also

- [isNull](#) in [PValue class](#).

## getDate

### Class

oracle.plsql.PDate

### Syntax

```
public void getDate(Date date)
```

### Purpose

Gets the value of this PDate.

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the Date value.

---

---

### Parameters

value - the date value to set to

### Throws

NullPointerException if the value is NULL.

### See Also

- [isNull](#) in [PValue class](#).

## setValue

### Class

oracle.plsql.PDate

### Syntax

```
public void setValue(Date date)
```

### Purpose

Sets the value of this PDate. To set this PDate to null, use [setNull\(\)](#).

### Parameters

value - the Date value to set to

### See Also

- [setNull](#) in [PValue class](#).

## toString

### Class

oracle.plsql.PDate

### Syntax

```
public String toString()
```

### Returns

The string representation of this PDate.

### Overrides

[toString](#) in [PValue](#) class.

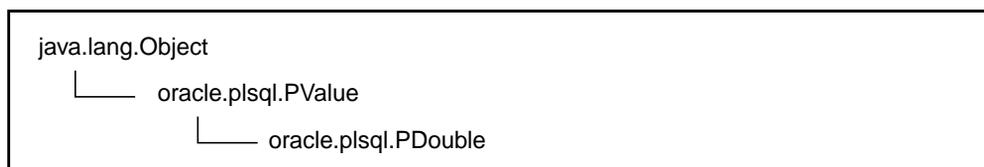
---

## PDouble class

```
public class PDouble
extends PValue
```

This class encapsulates PL/SQL numbers (real values).

*Figure 7-4 Inheritance for the PDouble class*



## PDouble constructor

### Class

oracle.plsql.PDouble

### Syntax

```
public PDouble()
```

```
public PDouble(double value)
```

```
public PDouble(int value)
```

### Purpose

Constructs a null PL/SQL number.

### Parameters

value - the initial value of the PDouble

## copy

### Class

oracle.plsql.PDouble

### Syntax

```
public void copy(PDouble value)
```

### Purpose

Copies the PDouble.

### Parameters

value - the PValue to copy from

## doubleValue

### Class

oracle.plsql.PDouble

### Syntax

```
public double doubleValue()
```

---

---

**Note:** You should use `isNull()` to determine if this value is NULL before getting the double value.

---

---

### Returns

The value of this PDouble as a double.

### Throws

NullPointerException if the value is NULL.

**See Also**

- [isNull](#) in [PValue class](#).

## intValue

**Class**

oracle.plsql.PDouble

**Syntax**

```
public int intValue()
```

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the integer value.

---

---

**Returns**

The value of this PDouble as an integer.

**Throws**

NullPointerException if the value is NULL.

**See Also**

- [isNull](#) in [PValue class](#).

## setValue

**Class**

oracle.plsql.PDouble

**Syntax**

```
public void setValue(double value)
```

```
public void setValue(int value)
```

### **Purpose**

Sets the value of this PDouble. To set this PDouble to null, use [setNull\(\)](#).

### **Parameters**

value - the double or integer value to set

### **See Also**

- [setNull](#) in [PValue class](#).

## **toString**

### **Class**

oracle.plsql.PDouble

### **Syntax**

```
public String toString()
```

### **Returns**

The string representation of this PDouble.

### **Overrides**

[toString](#) in [PValue class](#).

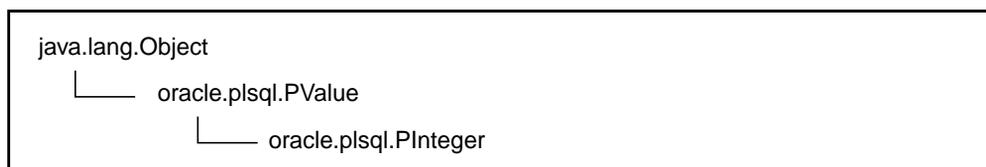
---

## PInteger class

```
public class PInteger  
extends PValue
```

This class encapsulates PL/SQL binary integers.

*Figure 7-5 Inheritance for the PInteger class*



## PInteger constructor

### Class

`oracle.plsql.PInteger`

### Syntax

```
public PInteger()
```

```
public PInteger(int value)
```

### Purpose

Constructs a NULL PL/SQL binary integer.

### Parameters

value - the initial value of the PInteger

## copy

### Class

oracle.plsql.PInteger

### Syntax

```
public void copy(PInteger value)
```

### Purpose

Copies a PInteger.

### Parameters

value - the PInteger to copy from

## intValue

### Class

oracle.plsql.PInteger

### Syntax

```
public int intValue()
```

---

---

**Note:** You should use `isNull()` to determine if this value is NULL before getting the integer value.

---

---

### Returns

The value of this PInteger as an integer.

### Throws

NullPointerException if the value is NULL.

**See Also**

- [isNull](#) in [PValue class](#).

**setValue****Class**

oracle.plsql.PInteger

**Syntax**

```
public void setValue(int value)
```

**Purpose**

Sets the value of this PInteger. To set this PInteger to null, use [setNull\(\)](#).

**Parameters**

value - the integer value to set

**See Also**

- [setNull](#) in [PValue class](#).

**toString****Class**

oracle.plsql.PInteger

**Syntax**

```
public String toString()
```

### Returns

The string representation of this PInteger.

### Overrides

[toString](#) in [PValue](#) class.

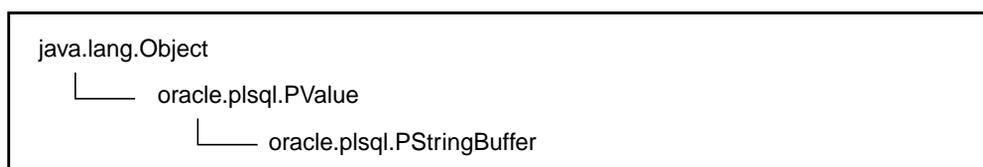
---

## PStringBuffer class

```
public class PStringBuffer  
extends PValue
```

This class encapsulates PL/SQL strings: char, varchar2 and long. It is implemented to look like `java.lang.StringBuffer` as closely as possible. However, one key difference is that the capacity of a `PStringBuffer` indicates the number of bytes, not Java chars, allocated to store the string as this `PStringBuffer` is used to store PL/SQL char, varchar2 and long. Another difference is that once created, a `PStringBuffer` can not grow dynamically. Also, the strings will be converted to the NLS character set currently selected by `NLS_LANG` environment property.

*Figure 7-6 Inheritance for the PStringBuffer class*



### See Also

- `StringBuffer`

## PStringBuffer constructor

### Class

```
oracle.plsql.PStringBuffer
```

### Syntax

```
public PStringBuffer(int capacity)
```

```
public PStringBuffer(String str)
```

```
public PStringBuffer(byte str[])
```

```
public PStringBuffer(  
    int          capacity,  
    String       str)
```

```
public PStringBuffer(  
    int          capacity,  
    byte[]       str[])
```

```
public PStringBuffer(  
    int          length,  
    StringBuffer str)
```

### Purpose

Constructs a PL/SQL StringBuffer of a specific capacity (bytes).

---

---

**Note:** When you construct a PStringBuffer, its capacity should be large enough to hold the data being retrieved from the database. An `ArrayIndexOutOfBoundsException` will be thrown if the PStringBuffer is not large enough.

---

---

### Parameters

capacity - the capacity of the PStringBuffer (bytes)

str - the initial value of the PStringBuffer

length - the current size of the string buffer

## byteArrayValue

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public byte[] byteArrayValue()
```

**Returns**

The value of this PStringBuffer as a byte array.

**Throws**

[NullValueException](#) if the value is NULL.

## capacity

**Class**

oracle.plsql.PStringBuffer

**Syntax**

```
public int capacity()
```

**Returns**

The capacity of this PStringBuffer (in terms of bytes).

## charArrayValue

**Class**

oracle.plsql.PStringBuffer

**Syntax**

```
public char[] charArrayValue()
```

**Returns**

The string in char array.

**Throws**

[NullValueException](#) If the value is NULL.

## clone

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public Object clone()
```

### Purpose

Clone this PStringBuffer.

### Overrides

clone in class PValue.

## copy

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public void copy(PStringBuffer value)
```

### Purpose

Copies a PStringBuffer.

### Parameters

value - the PStringBuffer to copy from

## getBytesArray

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public byte[] getBytesArray()
```

### Returns

The byte array buffer of this PStringBuffer.

### Throws

[NullPointerException](#) if the value is NULL.

## getStringBuffer

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public void getStringBuffer(StringBuffer str)
```

### Purpose

Extracts the value of this PStringBuffer and put it in a StringBuffer.

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the StringBuffer value.

---

---

### Parameters

str - the StringBuffer to store the value

### Throws

[NullPointerException](#) if the value is NULL.

### See Also

- [isNull](#) in [PValue class](#).

## getValue

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public int getValue(  
    byte          dst[],  
    int          dstBegin)
```

### Purpose

Gets the value of this PStringBuffer into a byte array.

### Parameters

dst - destination buffer to copy value to

dstBegin - the starting position of the destination buffer

### Returns

The number of bytes copied.

### Overrides

Abstract method [getValue](#) in [PValue class](#).

## length

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public int length()
```

### Returns

The length of this PStringBuffer.

## setValue

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public void setValue(StringBuffer value)
```

```
public void setValue(String value)
```

```
public void setValue(byte src[])
```

```
public void setValue(  
    byte          src[],  
    int           srcBegin,  
    int           srcLength)
```

### Purpose

Sets the value of this PStringBuffer with the content of a byte array. To set this PStringBuffer to null, either pass a null StringBuffer or use [setNull\(\)](#).

### Parameters

value - the StringBuffer value to set

src - source buffer to copy value to

srcBegin - the starting position of the buffer

srcLength - the length of the buffer to use

### Throws

ArrayIndexOutOfBoundsException if StringBuffer's length exceeds capacity.

### See Also

- [setNull](#) in [PValue](#) class.

## stringBufferValue

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public StringBuffer stringBufferValue()
```

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the StringBuffer value.

---

---

### Returns

The value of this PStringBuffer as a StringBuffer.

### Throws

NullPointerException if the value is NULL.

**See Also**

- [isNull](#) in [PValue](#) class.

## stringValue

**Class**

oracle.plsql.PStringBuffer

**Syntax**

```
public String stringValue()
```

---

---

**Note:** You should use [isNull\(\)](#) to determine if this value is NULL before getting the String value.

---

---

**Returns**

The value of this PStringBuffer as a String.

**Throws**

[NullValueException](#) if the value is NULL.

**See Also**

- [isNull](#) in [PValue](#) class.

## toCharArray

**Class**

oracle.plsql.PStringBuffer

**Syntax**

```
public char[] toCharArray()
```

### Returns

The string in char array.

### Throws

NullPointerException if the value is NULL.

## toString

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public String toString()
```

### Returns

The String representation of this PStringBuffer.

### Overrides

[toString](#) in [PValue](#) class.

## useBuffer

### Class

oracle.plsql.PStringBuffer

### Syntax

```
public void useBuffer(boolean use)
```

### Purpose

Specifies whether this buffer should be used directly in PL/SQL execution. Default is false. This may speed up the performance of PL/SQL execution by minimizing

memory copy. However, if the same PL/SQL procedure or function is used repeatedly with different buffers, it will perform better by not using this buffer. In such a case, the wrapper class will allocate its own buffer to serve all calls.

### **Parameters**

use - use this buffer or not

## PValue class

```
public abstract class PValue
extends Object
implements Cloneable
```

This class is the base class of all PL/SQL datatypes. It provides a way to handle null values in PL/SQL.

**Figure 7-7 Inheritance for the PValue class**



When users want to invoke the procedures and functions in a PL/SQL package, they can do so by calling the methods in corresponding Java wrapper class generated by the utility **pl2java**. It extracts information about a given PL/SQL package and produces the Java class wrapper for the package. For example, the PL/SQL package `foo`:

```
CREATE OR REPLACE PACKAGE foo AS
    function bar(foo_bar in varchar2) return binary_integer;
END foo;
```

will generate a Java class `Foo`:

```
public class Foo {
    public Foo(Session dbSession) { ... }
    public PInteger bar(PStringBuffer foo_bar) throws ServerException { ... }
}
```

To invoke the PL/SQL function `foo.bar`, the users should do the following:

1. Create a new database session.
2. Instantiate the `Foo` wrapper class.
3. Creates the PL/SQL values for the parameters of `bar`.
4. Invoke the method `bar` of the instance of `Foo`.

The following sample codes illustrate how this is done:

```

import oracle.rdbms.*; // import Oracle Java classes
import oracle.plsql.*;
public class Main {
    public static void Main(String args[]) throws ServerException {
        // creates a database session and logon
        Session dbSession = new Session("scott", "tiger", "HR_DB");
        // instantiates the Foo wrapper class
        Foo foo = new Foo(dbSession);
        // creates the PL/SQL value for the parameter of foo
        PStringBuffer foo_bar = new PStringBuffer("test value");
        // invoke the function
        PInteger i = foo.bar(foo_bar);
    }
}

```

PL/SQL data types mapping to Java data types:

PL/SQL data type	Java class
BINARY_INTEGER (+ NATURAL, POSITIVE)	<a href="#">PInteger class</a>
NUMBER (+ DEC, DECIMAL, DOUBLE PRECISION, FLOAT, INTEGER, INT, NUMERIC, REAL, SMALLINT)	<a href="#">PDouble class</a>
CHAR(n) (+ CHARACTER, STRING)	<a href="#">PStringBuffer class</a>
VARCHAR2(n) (+ VARCHAR)	<a href="#">PStringBuffer class</a>
LONG	<a href="#">PStringBuffer class</a>
RAW (n)	<a href="#">PBoolean class</a>
LONG RAW	<a href="#">PByteArray class</a>
BOOLEAN	<a href="#">PByteArray class</a>
DATE	<a href="#">PDate class</a>
PL/SQL table	Java array (not a class)

### See Also

- [Session class](#)
- [PInteger class](#)
- [PDouble class](#)
- [PStringBuffer class](#)

- [PByteArray class](#)
- [PBoolean class](#)
- [PDate class](#)

## clone

### Class

oracle.plsql.PValue

### Syntax

```
public Object clone()
```

### Returns

A clone of this PValue.

### Overrides

clone in Object class.

## copy

### Class

oracle.plsql.PValue

### Syntax

```
public void copy(PValue value)
```

### Purpose

Copies a PValue.

**Parameters**

value - the PValue to copy from

**isNull****Class**

oracle.plsql.PValue

**Syntax**

```
public boolean isNull()
```

**Returns**

A boolean indicating if this value is NULL.

**setNull****Class**

oracle.plsql.PValue

**Syntax**

```
public void setNull()
```

**Purpose**

Sets this value to NULL.

## toString

### Class

oracle.plsql.PValue

### Syntax

```
public String toString()
```

### Returns

A String representation of this value.

### Overrides

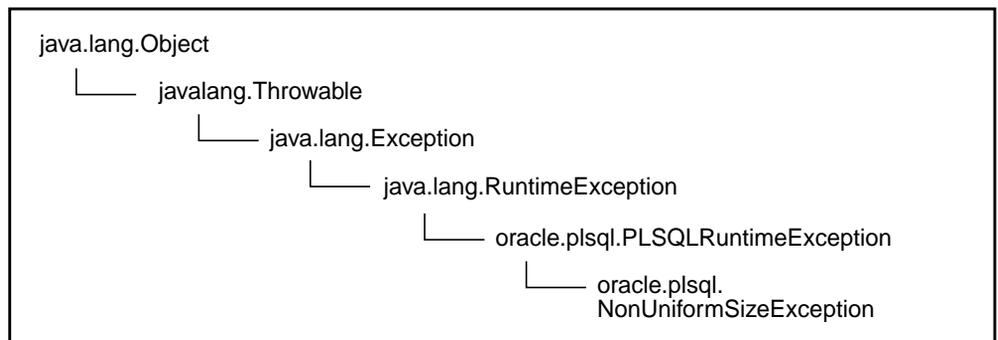
toString in Object class.

## NonUniformSizeException class

```
public class NonUniformSizeException
extends PLSQLRuntimeException
```

This class indicates an exception has occurred because the size of PL/SQL values in an array is not uniform.

**Figure 7–8 Inheritance for the NonUniformSizeException class**



## NonUniformSizeException constructor

### Class

`oracle.plsql.NonUniformSizeException`

### Syntax

```
public NonUniformSizeException(String s)
```

### Purpose

Creates a new `NonUniformSizeException` with the specified detail message

### Parameters

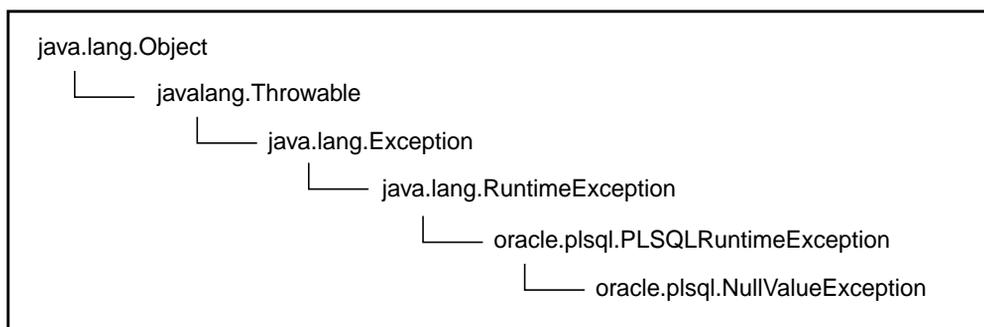
`s` - the detail message

## NullValueException class

**public class NullValueException**  
extends PLSQLRuntime Exception

This class indicates an exception has occurred because a PL/SQL NULL value is detected.

*Figure 7–9 Inheritance for the NullValueException class*



## NullValueException constructor

### Class

oracle.plsql.NullValueException

### Syntax

```
public NullValueException(String s)
```

### Purpose

Creates a new NullValueException with the specified detail message

### Parameters

s - the detail message

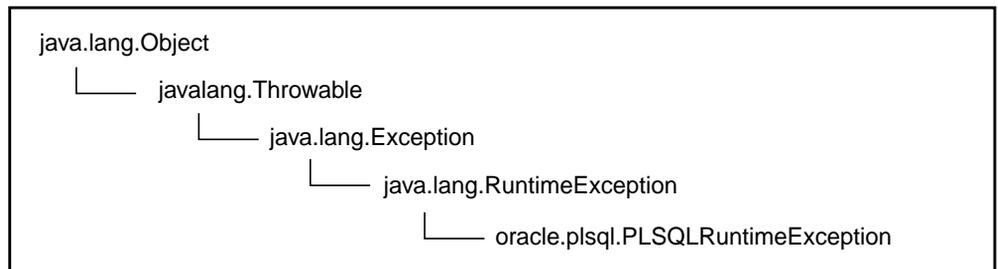
---

## PLSQLRuntimeException class

public class **PLSQLRuntimeException**  
extends RuntimeException

This class indicates an exception has occurred in PL/SQL at runtime.

**Figure 7-10 Inheritance for the PLSQLRuntimeException class**



## PLSQLRuntimeException constructor

### Class

`oracle.plsql.PLSQLRuntimeException`

### Syntax

```
public PLSQLRuntimeException(String s)
```

### Purpose

Creates a new `PLSQLRuntimeException` with the specified detail message.

### Parameters

`s` - the detail message



---

## oracle.rdbms package

This chapter contains reference pages for the following classes:

- Session class
  - Session constructor (does not log in)
  - Session constructor (logs in)
  - autoCommit
  - charToByteSize
  - commit
  - getMaxChars
  - getMaxCharsInArray
  - getMaxRows
  - setProperty, getProperty
  - getSessionInfo
  - getSessionProperty
  - isConnected
  - logon, logoff
  - rollback
  - setLogStream
  - setProperties
  - setSessionProperty

---

This chapter contains reference pages for the following exceptions:

- [Session class](#)
  - [ServerException constructor](#)
  - [getSqlcode](#)
  - [getSqlerrm](#)

---

## Session class

public class **Session**  
 extends Object

This class provides access to Oracle database: log on to and log off from Oracle database, and perform database operations.

*Figure 8–1 Inheritance for the Session class*



### Constants

Name	Syntax	Description
<b>globalAccess</b>	<code>public static final String globalAccess</code>	Session's global synchronization object. This is intended for internal use only

## Session constructor (does not log in)

### Class

oracle.rdbms.Session

### Syntax

`public Session()`

`public Session(Properties props)`

### Purpose

Creates a new database session but does not log on to that session.

## Parameters

props - the properties (e.g. ORACLE\_HOME) of this session

## See Also

- [logon, logoff](#)

## Session constructor (logs in)

### Class

oracle.rdbms.Session

### Syntax

```
public Session(String usernamePassword) throws ServerException
```

```
public Session(  
    String      usernamePassword,  
    Properties  props) throws ServerException
```

```
public Session(  
    String      username,  
    String      password,  
    String      connectStr) throws ServerException
```

```
public Session(  
    String      username,  
    String      password,  
    String      connectStr,  
    Properties  props) throws ServerException
```

### Purpose

Creates a new database session and logs on to that session.

### Parameters

username - the username used to logon

password - the user's password

connectStr - Oracle's connect string

props - the properties (e.g. ORACLE\_HOME) of this session

### Throws

[ServerException](#) if logon fails.

### Example

```
// Create a new database session and passe username, password
// and a connect-string to logon.
dbSession = new Session("rpang", "rpang", "wdk7322");
```

### See Also

- [logon, logoff](#)

## autoCommit

### Class

oracle.rdbms.Session

### Syntax

```
public void autoCommit(boolean autoCommit) throws ServerException
```

### Purpose

Specifies whether this session should perform commit immediately after each PL/SQL function call. The default is auto-commit off.

### Parameters

autoCommit - true if auto-commit, false if not

### Throws

[ServerException](#) if the method fails to change auto-commit property.

## charToByteSize

### Class

oracle.rdbms.Session

### Syntax

```
public int charToByteSize(int charCount)
```

### Purpose

Returns the size (in bytes) of a buffer needed to hold a given number of characters in the character set chosen in the current NLS\_LANG.

### Parameters

charCount - the number of characters

## commit

### Class

oracle.rdbms.Session

### Syntax

```
public void commit() throws ServerException
```

### Purpose

Commits database changes. This session will perform commit automatically by default when this session ends. This property can be changed by users.

### Throws

[ServerException](#) if commit fails.

### See Also

- [autoCommit](#)

## getMaxChars

### Class

oracle.rdbms.Session

### Syntax

```
public int getMaxChars() throws ServerException
```

```
public int getMaxChars(int maxChars) throws ServerException
```

### Purpose

Returns the maximum number of characters a PStringBuffer may store. This limit is imposed by RDBMS kernel and is dependent on the client and the database character sets. If *maxChars* is specified, the returned value will be limited by *maxChars*.

### Parameters

*maxChars* - the upper limit of chars

### Returns

The maximum number of characters a PStringBuffer may store.

### Throws

[ServerException](#) if the method fails to determine database character set.

## getMaxCharsInArray

### Class

oracle.rdbms.Session

### Syntax

```
public int getMaxCharsInArray(int rowCount) throws ServerException
```

```
public int getMaxCharsInArray(  

```

```
int          rowCount,  
int          maxChars) throws ServerException
```

### Purpose

Returns the maximum number of characters each PStringBuffer of a PStringBuffer array may store. This limit is imposed by RDBMS kernel and is dependent on the client and the database character sets. If *maxChars* is specified, the returned value will be limited by *maxChars*.

### Parameters

rowCount - number of rows in the PStringBuffer array

maxChars - the upper limit of chars

### Returns

The maximum number of characters each PStringBuffer of a PStringBuffer array may store.

### Throws

[ServerException](#) if the method fails to determine database character set.

## getMaxRows

### Class

oracle.rdbms.Session

### Syntax

```
public int getMaxRows(int rowSize) throws ServerException
```

```
public int getMaxRows(  
    int          rowSize,  
    int          maxRows) throws ServerException
```

**Purpose**

Returns the maximum number of rows of PStringBuffer a PStringBuffer array may have. This limit is imposed by RDBMS kernel and is dependent on the client and the database character sets. If *maxRows* is specified, the returned value will be limited by *maxRows*.

**Parameters**

rowSize - number of characters in each PStringBuffer

maxRows - the upper limit of the number of rows

**Returns**

The maximum number of rows of PStringBuffer a PStringBuffer array may have.

**Throws**

[ServerException](#) if fails to determine database character set.

## setProperty, getProperty

**Class**

oracle.rdbms.Session

**Syntax**

```
public static void setProperty(  
    String      name,  
    String      value)
```

```
public static String getProperty(String name)
```

**Purpose**

Sets or returns Oracle's global environment. For example, ORACLE\_HOME.

**Parameters**

name - property name

value - property value

### Example

```
// Defines Oracle Session ORACLE_HOME property. ORACLE_HOME value
// can be retrieved from Java's system properties if this class is
// executed in the JServlet cartridge.
Session.setProperty("ORACLE_HOME", System.getProperty("oracleHome"));
```

## getSessionInfo

### Class

oracle.rdbms.Session

### Syntax

```
public String getSessionInfo()
```

### Returns

The session information about the logon.

## getSessionProperty

### Class

oracle.rdbms.Session

### Syntax

```
public String getSessionProperty(String name) throws ServerException
```

### Purpose

Returns the NLS property of the current session. The property name should be in upper case.

**Parameters**

name - the name of the NLS session property

**Returns**

The NLS property, or null if the property is not defined.

**Throws**

[ServerException](#) if fails to retrieve session NLS property.

**isConnected****Class**

oracle.rdbms.Session

**Syntax**

```
public boolean isConnected() throws ServerException
```

**Purpose**

Checks if this session is connected or not.

**Returns**

`true` if this session is still connected, `false` otherwise.

**Throws**

[ServerException](#) if the method fails to detect connection.

**logon, logoff****Class**

oracle.rdbms.Session

## Syntax

```
public void logon(String usernamePassword) throws ServerException
```

```
public void logon(  
    String      username,  
    String      password,  
    String      connectStr) throws ServerException
```

```
public void logoff() throws ServerException
```

## Purpose

Either logs on to or logs off of a database.

## Parameters

usernamePassword - the username/password@connect-string used to logon

username - the username used to logon

password - the user's password

connectStr - Oracle's connect string

## Throws

[ServerException](#) if logon or logoff fails.

## rollback

### Class

oracle.rdbms.Session

### Syntax

```
public void rollback() throws ServerException
```

### Purpose

Rollback database changes.

**Throws**

ServerException if rollback fails.

**setLogStream****Class**

oracle.rdbms.Session

**Syntax**

```
public void setLogStream(OutputLogStream logStream)
```

**Purpose**

Sets the log stream.

**Parameters**

logStream - the log stream to use

**setProperties****Class**

oracle.rdbms.Session

**Syntax**

```
public void setProperties(boolean check)
```

**Purpose**

Set this session's properties in the environment. This is for internal use only.

### Parameters

check - if `true` then check if the environment properties have been changed by another session

## setSessionProperty

### Class

oracle.rdbms.Session

### Syntax

```
public void setSessionProperty(  
    String      name,  
    String      value) throws ServerException
```

### Purpose

Sets Oracle RDBMS Session environment. For example, `NLS_DATE_LANGUAGE`. This is done by issuing the SQL statement:

```
alter session set <name> = <value>
```

### Parameters

name - property name

value - property value

### Throws

[ServerException](#) if the method fails to set Session property.

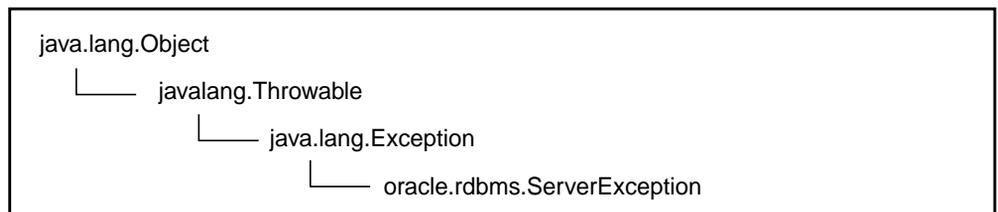
---

## ServerException class

```
public class ServerException  
extends Exception
```

This class encapsulates Oracle Server exceptions. It contains the SQL error code and error message.

*Figure 8–2 Inheritance for the ServerException class*



## ServerException constructor

### Class

oracle.rdbms.ServerException

### Syntax

```
public ServerException(  
    int          sqlcode,  
    String       sqlerm)
```

### Purpose

Constructs an Oracle Server exception with SQL code and error message.

### Parameters

sqlcode - the SQL code

sqlerr - the SQL error message

## getSqlcode

### Class

oracle.rdbms.ServerException

### Syntax

```
public int getSqlcode()
```

### Returns

The SQL error code.

## getSqlerrm

### Class

oracle.rdbms.ServerException

### Syntax

```
public String getSqlerrm()
```

### Returns

The SQL error message.

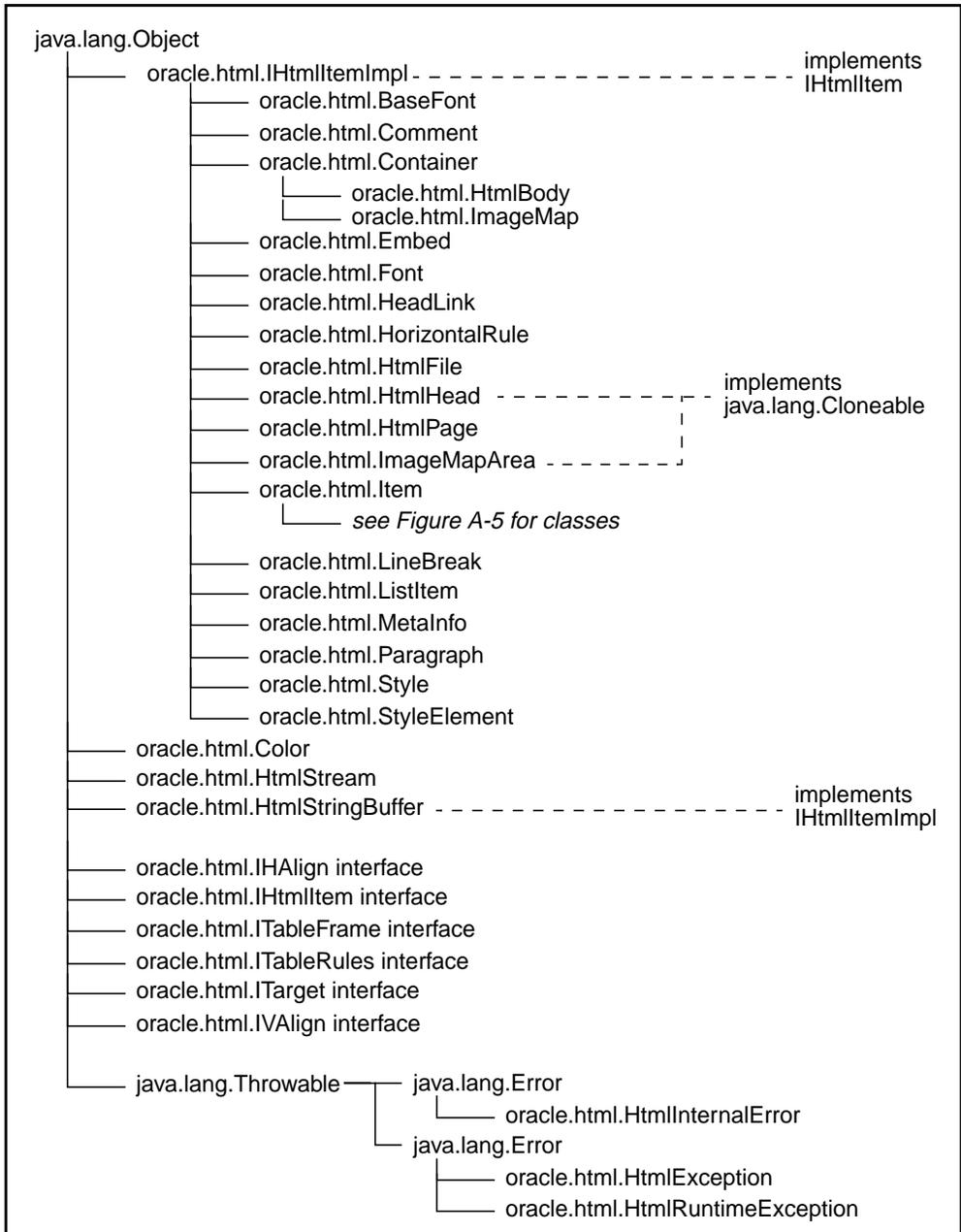
---

## Class Hierarchy

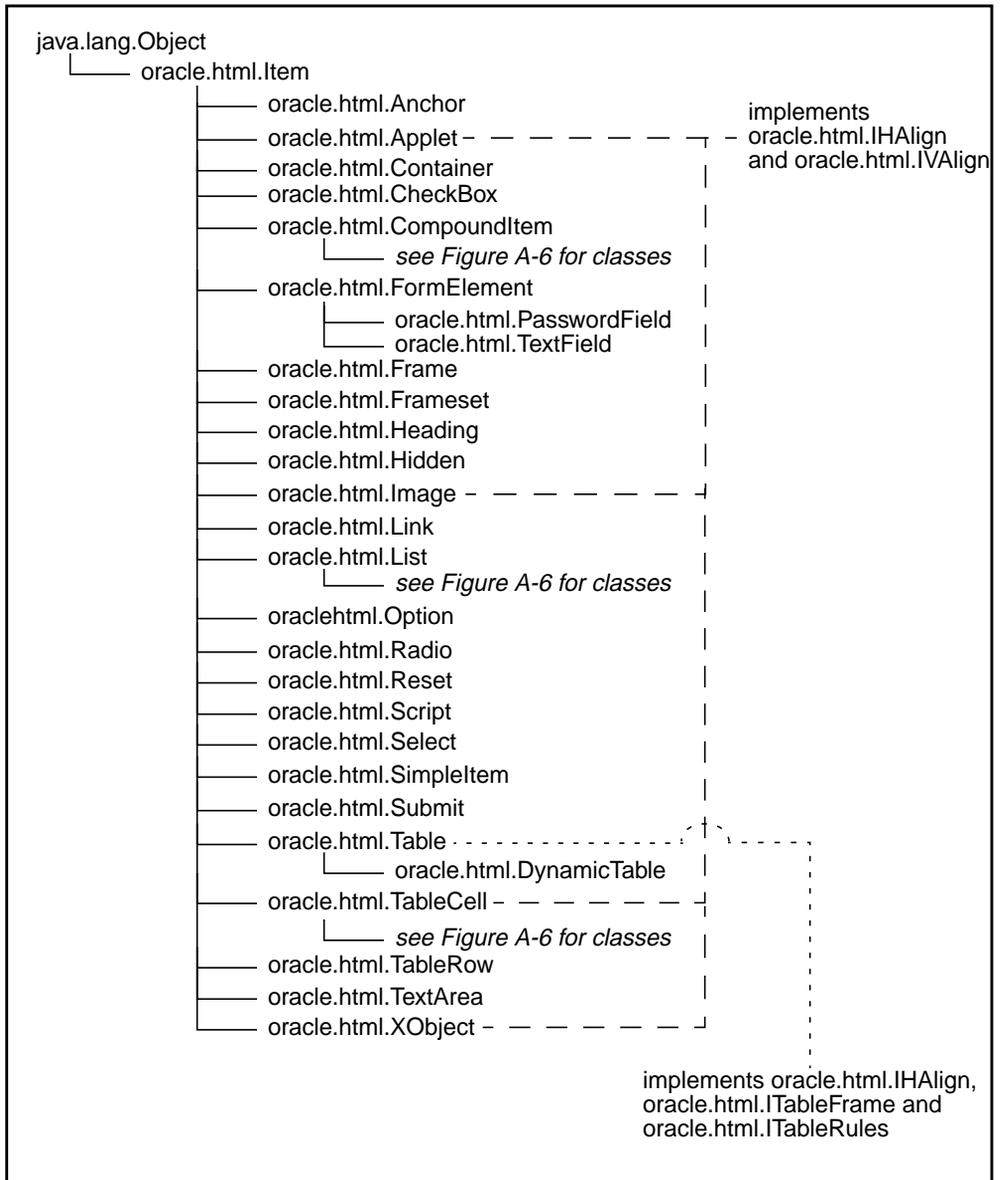
The following figures illustrate the class hierarchy for the JWeb Toolkit.

- Figure A-1, oracle.html package
- Figure A-2, oracle.html.Item class
- Figure A-3, Some oracle.html subclasses
- Figure A-4, oracle.OAS.servlet.http package
- Figure A-5, oracle.owas.wrb package
- Figure A-6, oracle.OAS.Services.ICX package
- Figure A-7, oracle.lang package
- Figure A-8, oracle.owas.nls package
- Figure A-9, oracle.plsql package
- Figure A-10, oracle.rdbms package

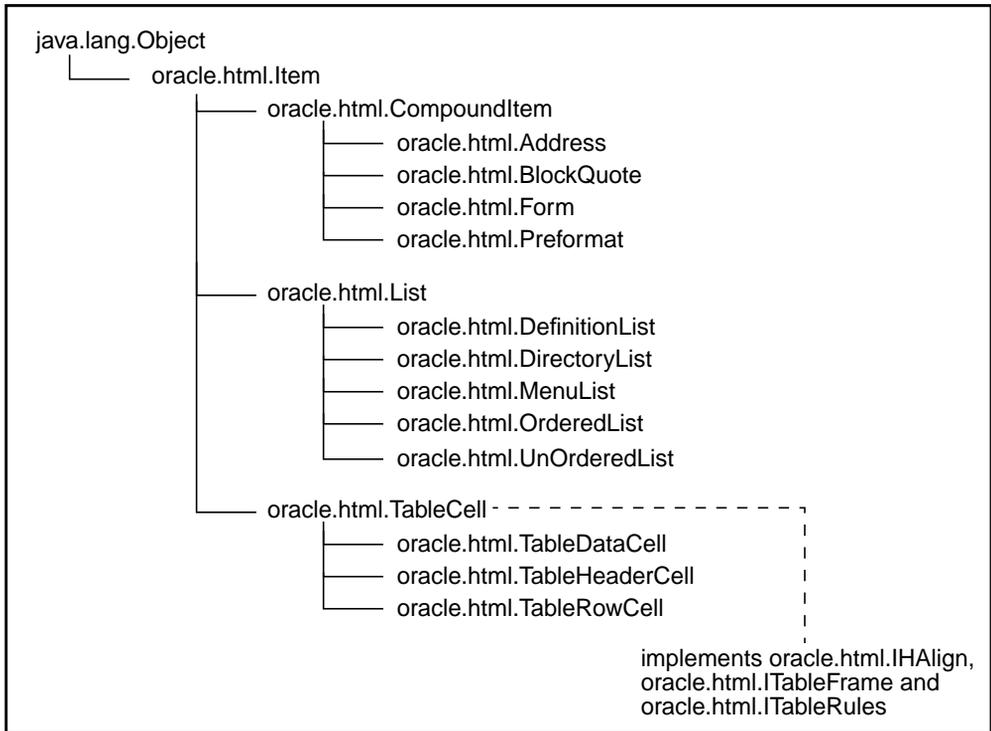
**Figure A-1** oracle.html package



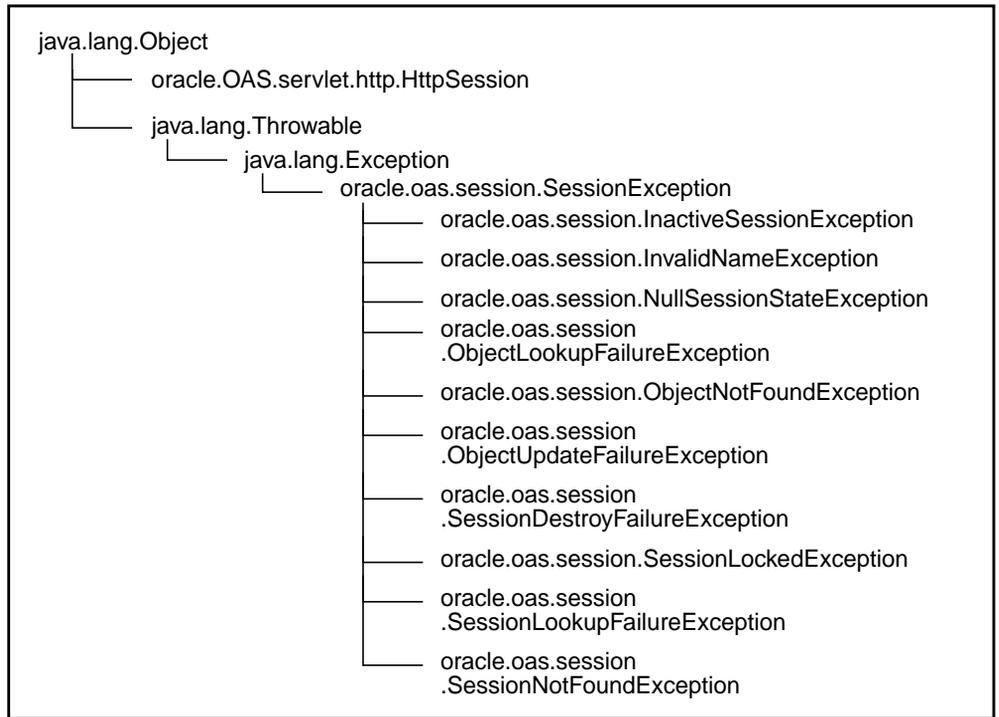
**Figure A-2** *oracle.html.Item class*



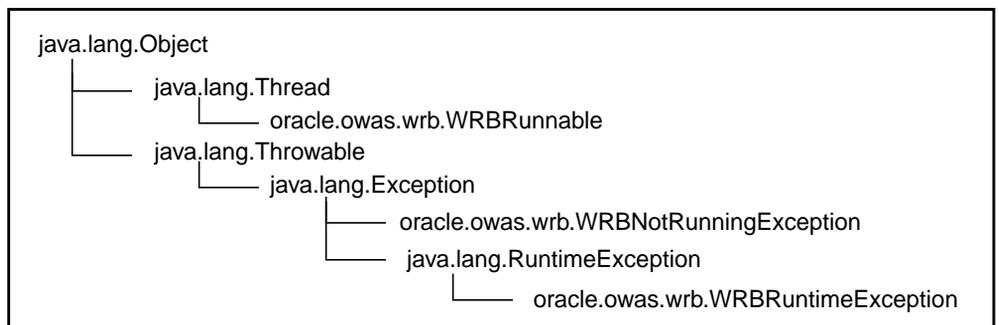
**Figure A-3** Some oracle.html subclasses



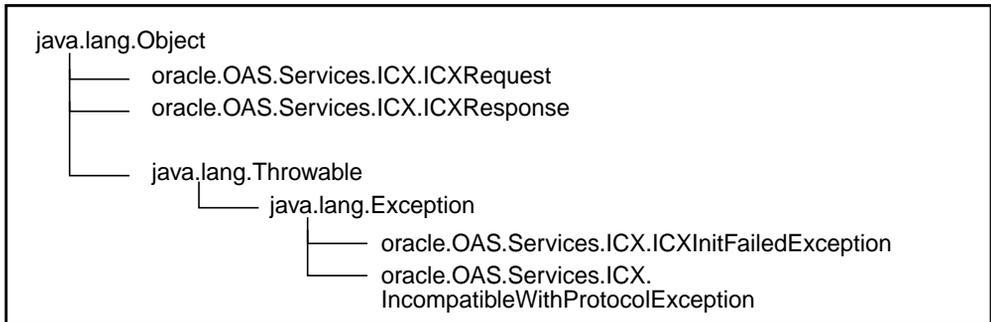
**Figure A-4** *oracle.OAS.servlet.http package*



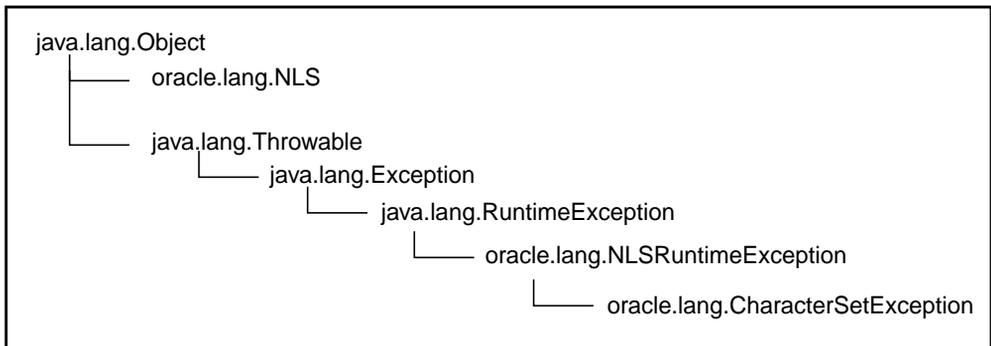
**Figure A-5** *oracle.owas.wrb package*



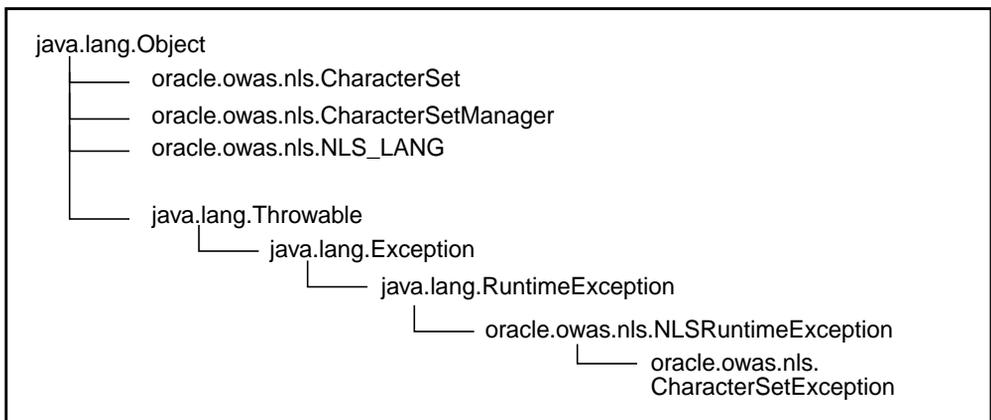
**Figure A-6** *oracle.OAS.Services.ICX package*



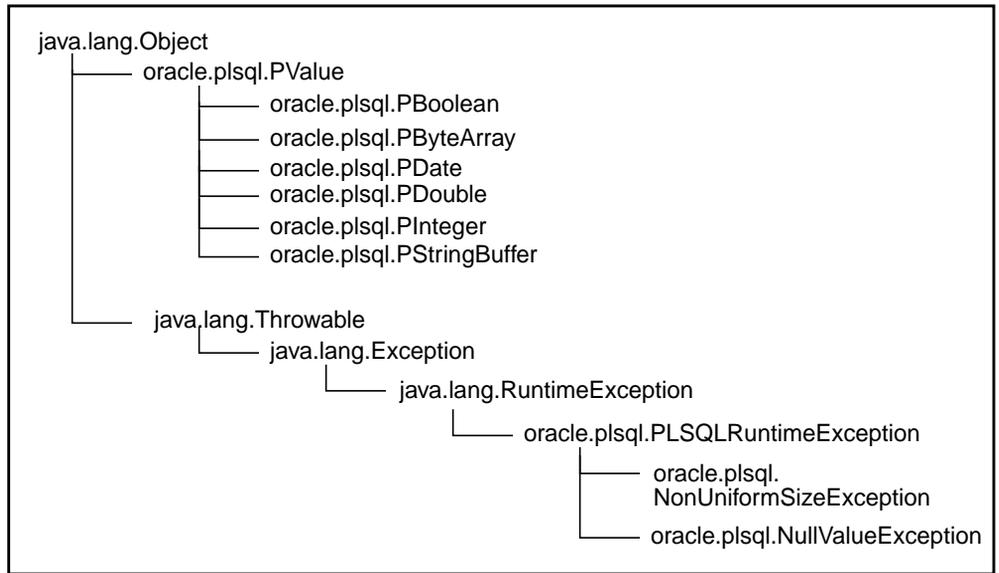
**Figure A-7** *oracle.lang package*



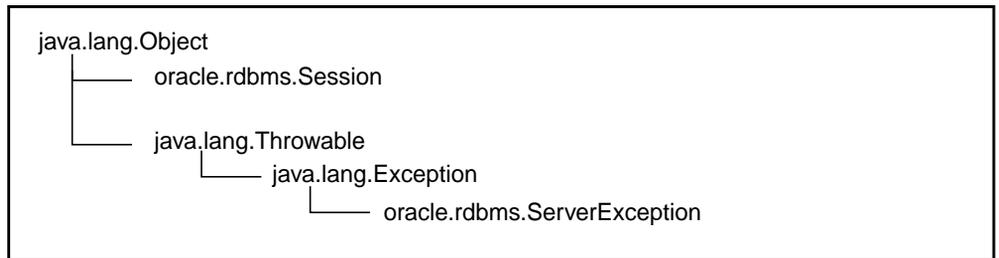
**Figure A-8** *oracle.owas.nls package*



**Figure A-9** *oracle.plsql package*



**Figure A-10** *oracle.rdbms package*





# B

---

## Locations of JServlet Class Files

---

Package	Description
oracle.html	Contains classes for generating HTML code. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>
oracle.OAS.servlet.http	Classes for programatic sessions. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jservlet.jar</b>
oracle.owas.wrb.WRB	Classes for accessing WRB services. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>
oracle.OAS.Services.ICX	Classes for sending ICX requests. Location: <b>\$ORAWEB_HOME/classes/services.jar</b>
oracle.lang	This package has been replaced by oracle.owas.nls. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>
oracle.owas.nls	Contains classes for setting and getting character sets. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>
oracle.plsql	Classes that map to PL/SQL data types. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>
oracle.rdbms	Classes for connecting to Oracle databases. Location: <b>\$ORAWEB_HOME/./cartx/jweb/classes/jweb.jar</b>

---



---

---

# Index

## A

---

ABOVE constant

    ITableFrame interface, 1-9

addArea

    ImageMap class, 1-155

addCell

    TableRow class, 1-263

addDef

    DefinitionList class, 1-57

addElement

    Style class, 1-228

addFrame

    Frameset class, 1-93

addItem

    Anchor class, 1-16

    CompoundItem class, 1-45

    Container class, 1-50

    HtmlBody class, 1-110

    HtmlPage class, 1-128

    Link class, 1-181

    List class, 1-185

    TableDataCell class, 1-255

addLink

    HtmlHead class, 1-120

addMetaInfo

    HtmlHead class, 1-120

addOption

    Select class, 1-220

addParam

    Applet class, 1-21

    Embed class, 1-67

    XObject class, 1-281

addProperty

    StyleElement class, 1-234

Address class, 1-13

Address constructor, 1-14

addRow

    DynamicTable class, 1-63

addStyle

    HtmlHead class, 1-121

ALL constant

    ITableRules interface, 1-10

Anchor class, 1-15

Anchor constructor, 1-16

append

    HtmlStringBuffer class, 1-141

appendln

    HtmlStringBuffer class, 1-143

Applet class, 1-20

Applet constructor, 1-21

aqua constant

    Color class, 1-36

ASCII constant

    NLS class, 5-2

ATTR\_ALGN\_CENTER constant

    Item class, 1-160

ATTR\_BIG constant

    Item class, 1-160

ATTR\_BOLD constant

    Item class, 1-160

ATTR\_CITATION constant

    Item class, 1-160

ATTR\_CODE constant

    Item class, 1-160

ATTR\_DEFINITION constant

    Item class, 1-160

ATTR\_EMPHASIS constant

- Item class, 1-160
- ATTR\_FONT\_BIG constant
  - Item class, 1-160
- ATTR\_FONT\_BOLD constant
  - Item class, 1-160
- ATTR\_FONT\_ITALIC constant
  - Item class, 1-160
- ATTR\_FONT\_SMALL constant
  - Item class, 1-160
- ATTR\_FONT\_STRIKE constant
  - Item class, 1-160
- ATTR\_FONT\_SUB constant
  - Item class, 1-160
- ATTR\_FONT\_SUPER constant
  - Item class, 1-160
- ATTR\_FONT\_TELETYPE constant
  - Item class, 1-161
- ATTR\_FONT\_UNDERLINE constant
  - Item class, 1-161
- ATTR\_ITALIC constant
  - Item class, 1-161
- ATTR\_KEYBOARD constant
  - Item class, 1-161
- ATTR\_PHRASE\_CITATION constant
  - Item class, 1-161
- ATTR\_PHRASE\_CODE constant
  - Item class, 1-161
- ATTR\_PHRASE\_DEFINITION constant
  - Item class, 1-161
- ATTR\_PHRASE\_EMPHASIS constant
  - Item class, 1-161
- ATTR\_PHRASE\_KEYBOARD constant
  - Item class, 1-161
- ATTR\_PHRASE\_SAMPLE constant
  - Item class, 1-161
- ATTR\_PHRASE\_STRONG constant
  - Item class, 1-161
- ATTR\_PHRASE\_VARIABLE constant
  - Item class, 1-161
- ATTR\_SAMPLE constant
  - Item class, 1-161
- ATTR\_SMALL constant
  - Item class, 1-161
- ATTR\_STRIKE constant
  - Item class, 1-161

- ATTR\_STRONG constant
  - Item class, 1-161
- ATTR\_SUB constant
  - Item class, 1-161
- ATTR\_SUPER constant
  - Item class, 1-161
- ATTR\_TELETYPE constant
  - Item class, 1-161
- ATTR\_UNDERLINE constant
  - Item class, 1-161
- ATTR\_VARIABLE constant
  - Item class, 1-161
- autoCommit
  - Session class, 8-5

## B

---

- backgroundColor
  - Table class, 1-243
  - TableCell class, 1-251
- backgroundImage
  - Table class, 1-243
- base variable
  - InvalidNameException class, 2-12
  - ObjectLookupFailureException class, 2-15
  - ObjectNotFoundException class, 2-17
  - ObjectUpdateFailureException class, 2-19
  - SessionDestroyFailureException class, 2-21
  - SessionException class, 2-23
  - SessionLockedException class, 2-25
  - SessionLookupFailureException class, 2-27
  - SessionNotFoundException class, 2-29
- BaseFont class, 1-28
- BaseFont constructor, 1-28
- BASELINE constant
  - IAlign interface, 1-12
- BASIC constant
  - ITableRules interface, 1-10
- BELOW constant
  - ITableFrame interface, 1-9
- black constant
  - Color class, 1-36
- BLANK constant
  - ITarget interface, 1-11
- BlockQuote class, 1-31

- BlockQuote constructor, 1-32
- blue constant
  - Color class, 1-36
- booleanValue
  - PBoolean class, 7-5
- border
  - Table class, 1-244
- BORDER constant
  - ITableFrame interface, 1-9
- borderColor
  - Table class, 1-244
- BOTTOM constant
  - IAlign interface, 1-12
- BOX constant
  - ITableFrame interface, 1-9
- byteArrayValue
  - PByteArray class, 7-9
  - PStringBuffer class, 7-28
- bytesToBytes
  - CharacterSet class, 6-4
- bytesToChars
  - CharacterSet class, 6-5
- bytesToString
  - CharacterSet class, 6-6

## C

---

- capacity
  - PStringBuffer class, 7-29
- caption
  - Table class, 1-245
- captionAlign
  - Table class, 1-245
- cellAt
  - TableRow class, 1-264
- cellPadding
  - Table class, 1-246
- cells
  - TableRow class, 1-264
- cellSpacing
  - Table class, 1-246
- CENTER constant
  - IAlign interface, 1-6
- CharacterSet class, 6-3
- CharacterSet constructor, 6-4

- CharacterSetException class
  - oracle.lang package, 5-7
  - oracle.owas.nls package, 6-25
- CharacterSetException constructor
  - oracle.lang package, 5-7
  - oracle.owas.nls package, 6-25
- CharacterSetManager class, 6-13
- charArrayValue
  - PStringBuffer class, 7-29
- charsToBytes
  - CharacterSet class, 6-7
- charToByteSize
  - Session class, 8-6
- CheckBox class, 1-33
- CheckBox constructor, 1-33
- class files, B-1
- clearAttr
  - Item class, 1-163
- clone
  - HtmlFile class, 1-116
  - HtmlPage class, 1-128
  - PByteArray class, 7-9
  - PDate class, 7-15
  - PStringBuffer class, 7-30
  - PValue class, 7-40
- Color class, 1-36
- Color constructor, 1-38
- COLS constant
  - ITableRules interface, 1-10
- colSpan
  - TableDataCell class, 1-256
- Comment class, 1-40
- Comment constructor, 1-41
- commit
  - Session class, 8-6
- CompoundItem class, 1-43
- CompoundItem constructor, 1-44
- connect
  - ICXRequest class, 4-4
- constants
  - ABOVE
    - ITableFrame interface, 1-9
  - ALL
    - ITableRules interface, 1-10
  - ASCII

NLS class, 5-2  
 ATTR\_ALGN\_CENTER  
   Item class, 1-160  
 ATTR\_BIG  
   Item class, 1-160  
 ATTR\_BOLD  
   Item class, 1-160  
 ATTR\_CITATION  
   Item class, 1-160  
 ATTR\_CODE  
   Item class, 1-160  
 ATTR\_DEFINITION  
   Item class, 1-160  
 ATTR\_EMPHASIS  
   Item class, 1-160  
 ATTR\_FONT\_BIG  
   Item class, 1-160  
 ATTR\_FONT\_BOLD  
   Item class, 1-160  
 ATTR\_FONT\_ITALIC  
   Item class, 1-160  
 ATTR\_FONT\_SMALL  
   Item class, 1-160  
 ATTR\_FONT\_STRIKE  
   Item class, 1-160  
 ATTR\_FONT\_SUB  
   Item class, 1-160  
 ATTR\_FONT\_SUPER  
   Item class, 1-160  
 ATTR\_FONT\_TELETYPE  
   Item class, 1-161  
 ATTR\_FONT\_UNDERLINE  
   Item class, 1-161  
 ATTR\_ITALIC  
   Item class, 1-161  
 ATTR\_KEYBOARD  
   Item class, 1-161  
 ATTR\_PHRASE\_CITATION  
   Item class, 1-161  
 ATTR\_PHRASE\_CODE  
   Item class, 1-161  
 ATTR\_PHRASE\_DEFINITION  
   Item class, 1-161  
 ATTR\_PHRASE\_EMPHASIS  
   Item class, 1-161  
 ATTR\_PHRASE\_KEYBOARD  
   Item class, 1-161  
 ATTR\_PHRASE\_SAMPLE  
   Item class, 1-161  
 ATTR\_PHRASE\_STRONG  
   Item class, 1-161  
 ATTR\_PHRASE\_VARIABLE  
   Item class, 1-161  
 ATTR\_SAMPLE  
   Item class, 1-161  
 ATTR\_SMALL  
   Item class, 1-161  
 ATTR\_STRIKE  
   Item class, 1-161  
 ATTR\_STRONG  
   Item class, 1-161  
 ATTR\_SUB  
   Item class, 1-161  
 ATTR\_SUPER  
   Item class, 1-161  
 ATTR\_TELETYPE  
   Item class, 1-161  
 ATTR\_UNDERLINE  
   Item class, 1-161  
 ATTR\_VARIABLE  
   Item class, 1-161  
 BASELINE  
   IVAlign interface, 1-12  
 BASIC  
   ITableRules interface, 1-10  
 BELOW  
   ITableFrame interface, 1-9  
 BLANK  
   ITarget interface, 1-11  
 BORDER  
   ITableFrame, 1-9  
 BOTTOM  
   IVAlign interface, 1-12  
 BOX  
   ITableFrame interface, 1-9  
 CENTER  
   IAlign interface, 1-6  
 colors, 1-36 to 1-37  
 COLS  
   ITableRules interface, 1-10

DEFAULT-NLS\_LANG  
     NLS\_LANG class, 6-15  
 DEFINITION  
     List class, 1-183  
 DIRECTORY  
     List class, 1-183  
 FILE\_DOC  
     HtmlPage class, 1-125  
 FRAME\_DOC  
     HtmlPage class, 1-125  
 globalAccess  
     Session class, 8-3  
 HorizontalRule  
     SimpleItem class, 1-223  
 HSIDES  
     ITableFrame interface, 1-9  
 IANA\_CHARSET  
     CharacterSetManager class, 6-13  
 INITIAL\_CAPACITY  
     CompoundItem class, 1-43  
 LEFT  
     IAlign interface, 1-6  
 LHS  
     ITableFrame interface, 1-9  
 LineBreak  
     SimpleItem class, 1-223  
 MAX  
     IAlign interface, 1-6  
     IAlign interface, 1-12  
 MENU  
     List class, 1-183  
 MIDDLE  
     IAlign interface, 1-12  
 MIN  
     IAlign interface, 1-6  
     IAlign interface, 1-12  
 NONE  
     IAlign interface, 1-6  
     ITableRules interface, 1-10  
     IAlign interface, 1-12  
 NORMAL\_DOC  
     HtmlPage class, 1-125  
 ORACLE\_CHARSET  
     CharacterSetManager class, 6-13  
 ORDERED  
     List class, 1-183  
 Paragraph  
     SimpleItem class, 1-223  
 PARENT  
     ITarget interface, 1-11  
 RHS  
     ITableFrame interface, 1-9  
 RIGHT  
     IAlign interface, 1-6  
 ROWS  
     ITableRules interface, 1-10  
 SELF  
     ITarget interface, 1-11  
 Str  
     IAlign interface, 1-6  
     ITableFrame interface, 1-9  
     ITableRules interface, 1-10  
     ITarget interface, 1-11  
     IAlign interface, 1-12  
 TOP  
     ITarget interface, 1-11  
     IAlign interface, 1-12  
 TYPE\_CHECKBOX  
     FormElement class, 1-78  
 TYPE\_FILE  
     FormElement class, 1-78  
 TYPE\_HIDDEN  
     FormElement class, 1-78  
 TYPE\_IMAGE  
     FormElement class, 1-78  
 TYPE\_PASSWORD  
     FormElement class, 1-78  
 TYPE\_RADIO  
     FormElement class, 1-78  
 TYPE\_RESET  
     FormElement class, 1-78  
 TYPE\_SUBMIT  
     FormElement class, 1-78  
 TYPE\_TEXTFIELD  
     FormElement class, 1-78  
 UNORDERED  
     List class, 1-183  
 VOLD  
     ITableFrame interface, 1-9  
 VSIDES

- ITableFrame interface, 1-9
- constructors
  - Address, 1-14
  - Anchor, 1-16
  - Applet, 1-21
  - BaseFont, 1-28
  - BlockQuote, 1-32
  - CharacterSet, 6-4
  - CharacterSetException
    - oracle.lang package, 5-7
    - oracle.owas.nls package, 6-25
  - CheckBox, 1-33
  - Color, 1-38
  - Comment, 1-41
  - CompoundItem, 1-44
  - Container, 1-50
  - DefinitionList, 1-56
  - DirectoryList, 1-60
  - DynamicTable, 1-62
- errors
  - HtmlInternalError, 1-293
- exceptions
  - CharacterSetException
    - oracle.lang package, 5-7
    - oracle.owas.nls package, 6-25
  - HtmlException, 1-291
  - HtmlRuntimeException, 1-292
  - ICXInitFailedException, 4-15
  - InactiveSessionException, 2-11
  - IncompatibleWithProtocolException, 4-16
  - InvalidNameException, 2-12
  - NLSRuntimeException
    - oracle.lang package, 5-6
    - oracle.owas.nls package, 6-26
  - NonUniformSizeException, 7-43
  - NullSessionStateException, 2-14
  - NullValueException, 7-44
  - ObjectLookupFailureException, 2-15
  - ObjectNotFoundException, 2-17
  - ObjectUpdateFailureException, 2-19
  - PLSQLRuntimeException, 7-45
  - ServerException, 8-15
  - SessionDestroyFailureException, 2-21
  - SessionException, 2-23
  - SessionLockedException, 2-25
  - SessionLookupFailureException, 2-27
  - SessionNotFoundException, 2-29
  - WRBNotRunningException, 3-5
  - WRBRuntimeException, 3-6
- Font, 1-72
- Form, 1-76
- FormElement, 1-79
- Frame, 1-87
- Frameset, 1-92
- Heading, 1-100
- Hidden, 1-104
- HorizontalRule, 1-106
- HtmlBody, 1-109
- HtmlException, 1-291
- HtmlFile, 1-115
- HtmlHead, 1-119
- HtmlInternalError, 1-293
- HtmlPage, 1-126
- HtmlRuntimeException, 1-292
- HtmlStringBuffer, 1-140
- ICXInitFailedException, 4-15
- ICXRequest, 4-4
- ICXResponse, 4-10
- IHtmlItemImpl, 1-146
- Image, 1-150
- ImageMap, 1-155
- ImageMapArea, 1-157
- InactiveSessionException, 2-11
- IncompatibleWithProtocolException, 4-16
- InvalidNameException, 2-12
- Item, 1-162
- LineBreak, 1-177
- Link, 1-180
- List, 1-184
- ListItem, 1-190
- MenuList, 1-193
- NLS\_LANG, 6-15
- NLSRuntimeException
  - oracle.lang package, 5-6
  - oracle.owas.nls package, 6-26
- NonUniformSizeException, 7-43
- NullSessionStateException, 2-14
- NullValueException, 7-44
- ObjectLookupFailureException, 2-15
- ObjectNotFoundException, 2-17

- ObjectUpdateFailureException, 2-19
- Option, 1-198
- OrderedList, 1-201
- Paragraph, 1-202
- PasswordField, 1-206
- PBoolean, 7-4
- PDate, 7-14
- PDouble, 7-19
- PInteger, 7-23
- PLSQLRuntimeException, 7-45
- Preformat, 1-208
- PStringBuffer, 7-27
- Reset, 1-212
- Script, 1-216
- Select, 1-220
- ServerException, 8-15
- Session
  - does not log in, 8-3
  - logs in, 8-4
- SessionDestroyFailureException, 2-21
- SessionException, 2-23
- SessionLockedException, 2-25
- SessionLookupFailureException, 2-27
- SessionNotFoundException, 2-29
- StyleElement, 1-234
- Submit, 1-239
- Table, 1-242
- TableCell, 1-250
- TableDataCell, 1-254
- TableHeaderCell, 1-261
- TableRow, 1-263
- TableRowCell, 1-269
- TextArea, 1-272
- TextField, 1-276
- WRBNotRunningException, 3-5
- WRBRunnable, 3-2
- WRBRuntimeException, 3-6
- XObject, 1-280
- Container class, 1-49
- Container constructor, 1-50
- copy
  - PBoolean class, 7-5
  - PByteArray class, 7-10
  - PDate class, 7-15
  - PDouble class, 7-20

- PInteger class, 7-24
- PStringBuffer class, 7-30
- PValue class, 7-40
- currentCharSet
  - NLS class, 5-2
- cyan constant
  - Color class, 1-36

## D

---

- darkGray constant
  - Color class, 1-36
- dateValue
  - PDate class, 7-16
- DEFAULT-NLS-LANG constant
  - NLS\_LANG class, 6-15
- DEFINITION constant
  - List class, 1-183
- DefinitionList class, 1-55
- DefinitionList constructor, 1-56
- DIRECTORY constant
  - List class, 1-183
- DirectoryList class, 1-59
- DirectoryList constructor, 1-60
- disableTransaction
  - ICXRequest class, 4-5
- DOUBLE-QUOTE-CHAR variable
  - HtmlStringBuffer class, 1-140
- doubleValue
  - PDouble class, 7-20
- DynamicTable constructor, 1-62

## E

---

- elementAt
  - Style class, 1-229
- elements
  - Style class, 1-230
- Embed class, 1-66
- Embed constructor
  - constructors
    - Embed, 1-67
- enableTransaction
  - ICXRequest class, 4-5
- endTag

- Frame class, 1-88
- Frameset class, 1-93
- Table class, 1-247
- equals
  - CharacterSet class, 6-8
- error classes
  - HtmlInternalError, 1-293
- examples
- ICX, 4-3
- exception classes
  - CharacterSetException
    - oracle.lang package, 5-7
  - HtmlException, 1-291
  - HtmlRuntimeException, 1-292
  - ICXInitFailedException, 4-15
  - InactiveSessionException, 2-11
  - IncompatibleWithProtocolException, 4-16
  - InvalidNameSessionException, 2-12
  - NLSRuntimeException
    - oracle.lang package, 5-6
    - oracle.owas.nls package, 6-26
  - NonUniformSizeException, 7-43
  - NullSessionStateException, 2-14
  - NullValueException, 7-44
  - ObjectLookupFailureException, 2-15
  - ObjectNotFoundException, 2-17
  - ObjectUpdateFailureException, 2-19
  - PLSQLRuntimeException, 7-45
  - ServerException, 8-15
  - SessionDestroyFailureException, 2-21
  - SessionException, 2-23
  - SessionLockedException, 2-25
  - SessionLookupFailureException, 2-27
  - SessionNotFoundException, 2-29
  - WRBNotRunningException, 3-5
  - WRBRuntimeException, 3-6
- exception classes CharacterSetException
  - oracle.owas.nls package, 6-25

## F

---

- FILE\_DOC constant
  - HtmlPage class, 1-125
- finalize
  - ICXResponse class, 4-10

- Font class, 1-71
- Font constructor, 1-72
- Form class, 1-74
- Form constructor, 1-76
- FormElement class, 1-78
- FormElement constructor, 1-79
- frame
  - Table class, 1-247
- Frame class, 1-86
- Frame constructor, 1-87
- FRAME\_DOC constant
  - HtmlPage class, 1-125
- Frameset class, 1-91
- Frameset constructor, 1-92
- fuchsia constant
  - Color class, 1-36

## G

---

- getBody
  - HtmlPage class, 1-129
- getByteArray
  - PByteArray class, 7-10
  - PStringBuffer class, 7-31
- getChar
  - HtmlStringBuffer class, 1-143
- getCharacterSet
  - CharacterSetManager class, 6-13
  - NLS\_LANG class, 6-16
- getContent
  - ICXResponse class, 4-11
- getContentLanguage
  - NLS\_LANG class, 6-16
- getCreationTime
  - HttpSession class, 2-4
- getCurrent
  - NLS\_LANG class, 6-17
- getDate
  - PDate class, 7-16
- getHead
  - HtmlPage class, 1-129
- getHeader
  - ICXResponse class, 4-11
- getHeaders
  - ICXResponse class, 4-12

- getHTTPVersion
  - ICXResponse class, 4-12
- getIANACharacterSet
  - NLS\_LANG class, 6-17
- getId
  - HttpSession class, 2-5
- getISOLanguage
  - NLS\_LANG class, 6-18
- getISOTerritory
  - NLS\_LANG class, 6-18
- getLanguage
  - NLS\_LANG class, 6-19
- getLastAccessedTime
  - HttpSession class, 2-5
- getMaxBytesPerChar
  - CharacterSet class, 6-8
- getMaxChars
  - Session class, 8-7
- getMaxCharsInArray
  - Session class, 8-7
- getMaxInactiveInterval
  - HttpSession class, 2-6
- getMaxRows
  - Session class, 8-8
- getName
  - CharacterSet class, 6-9
- getNumRows
  - DynamicTable class, 1-63
- getOracleCharacterSet
  - NLS\_LANG class, 6-19
- getOracleLanguage
  - NLS\_LANG class, 6-20
- getOracleTerritory
  - NLS\_LANG class, 6-20
- getProperty
  - Session class, 8-9
- getRealm
  - ICXResponse class, 4-12
- getReasonPhrase
  - ICXResponse class, 4-13
- getSession
  - ServletRequest class, 2-2
- getSessionInfo
  - Session class, 8-10
- getSessionProperty

- Session class, 8-10
- getSqlcode
  - ServerException class, 8-16
- getSqlerrm
  - ServerException class, 8-16
- getStatusCode
  - ICXResponse class, 4-13
- getStringBuffer
  - PStringBuffer class, 7-31
- getTerritory
  - NLS\_LANG class, 6-21
- getValue
  - HttpSession class, 2-6
  - NLS\_LANG class, 6-21
  - PStringBuffer class, 7-32
- getValueNames
  - HttpSession class, 2-7
- getWRBContext
  - WRBRunnable class, 3-3
- globalAccess constant
  - Session class, 8-3
- gray constant
  - Color class, 1-36
- green constant
  - Color class, 1-36

## H

---

- hAlign
  - Table class, 1-247
- halign
  - TableCell class, 1-251
- Heading class, 1-99
- Heading constructor, 1-100
- HeadLink class, 1-96
- height
  - TableDataCell class, 1-256
- Hidden class, 1-103
- Hidden constructor, 1-104
- HorizontalRule class, 1-105
- HorizontalRule constant
  - SimpleItem class, 1-223
- HorizontalRule constructor, 1-106
- HSIDES constant
- ITableFrame interface, 1-9

## HTML elements

### mapping to Java classes

- body elements, 1-1
- form elements, 1-4
- head elements, 1-1
- list elements, 1-2
- other elements, 1-4
- table elements, 1-3
- text-level elements, 1-3

### html package, B-1

### HtmlBody class, 1-108

### HtmlBody constructor, 1-109

### HtmlException class, 1-291

### HtmlException constructor, 1-291

### HtmlFile class, 1-114

### HtmlFile constructor, 1-115

### HtmlHead class, 1-118

### HtmlHead constructor, 1-119

### HtmlInternalServerError class, 1-293

### HtmlInternalServerError constructor, 1-293

### HtmlPage class, 1-125

### HtmlPage constructors, 1-126

### HtmlRuntimeException class, 1-292

### HtmlRuntimeException constructor, 1-292

### HtmlStream class, 1-135

### HtmlStringBuffer class, 1-140

### HtmlStringBuffer constructor, 1-140

### http package, B-1

### HttpSession class, 2-4

## I

---

### IANA\_CHARSET constant

- CharacterSetManager class, 6-13

### ICX package, B-1

### ICXInitFailedException class, 4-15

### ICXInitFailedException constructor, 4-15

### ICXRequest class, 4-3

### ICXRequest constructor, 4-4

### ICXResponse class, 4-10

### ICXResponse constructor, 4-10

### IHAlign interface, 1-6

### IHtmlItem interface, 1-7

### IHtmlItemImpl class, 1-146

### IHtmlItemImpl constructor, 1-146

### Image class, 1-149

### Image constructor, 1-150

### ImageMap, 1-155

### ImageMap class, 1-154

### ImageMap constructor, 1-155

### ImageMapArea class, 1-157

### ImageMapArea constructor, 1-157

### InactiveSessionException class, 2-11

### InactiveSessionException constructor, 2-11

### IncompatibleWithProtocolException class, 4-16

### IncompatibleWithProtocolException

- constructor, 4-16

### INITIAL\_CAPACITY

- CompoundItem class, 1-43

### intValue

- PDouble class, 7-21

- PInteger class, 7-24

### invalidate

- HttpSession class, 2-7

### InvalidNameException class, 2-12

### InvalidNameException constructor, 2-12

### isConnected

- Session class, 8-11

### isNew

- HttpSession class, 2-8

### isNull

- PValue class, 7-41

### ITableFrame interface, 1-9

### ITableRules interface, 1-10

### ITarget interface, 1-11

### Item class, 1-160

### Item constructor, 1-162

### itemAt

- CompoundItem class, 1-45

- Container class, 1-51

- List class, 1-186

### items

- CompoundItem class, 1-46

- Container class, 1-52

### IValign interface, 1-12

## J

---

### jar files, B-1

## L

---

- lang package, B-1
- LEFT constant
  - IHAlign interface, 1-6
- length
  - HtmlStringBuffer class, 1-144
  - NLS class, 5-3
  - PByteArray class, 7-11
  - PStringBuffer class, 7-33
- LHS constant
  - ITableFrame interface, 1-9
- lightGray constant
  - Color class, 1-36
- lime constant
  - Color class, 1-36
- LineBreak class, 1-177
- LineBreak constant
  - SimpleItem class, 1-223
- LineBreak constructor, 1-177
- Link class, 1-179
- Link constructor, 1-180
- List constructor, 1-184
- ListItem class, 1-189
- ListItem constructor, 1-190
- location of class files, B-1
- logoff
  - Session class, 8-11
- logon
  - Session class, 8-11

## M

---

- magenta constant
  - Color class, 1-37
- maroon constant
  - Color class, 1-37
- MAX constant
  - IHAlign interface, 1-6
  - IValign interface, 1-12
- MENU constant
  - List class, 1-183
- MenuList constructor, 1-193
- MetaInfo class, 1-194
- MIDDLE constant

- IValign interface, 1-12
- MIN constant
  - IHAlign interface, 1-6
  - IValign interface, 1-12
- minByteSize
  - CharacterSet class
    - Java character strings, 6-9
    - other character sets, 6-10
- minCharSize
  - CharacterSet class, 6-11

## N

---

- name
  - Anchor class, 1-17
- navy constant
  - Color class, 1-37
- NEWLINE\_CHAR variable
  - HtmlStringBuffer class, 1-140
- NLS class, 5-2
- nls package, B-1
- NLS\_LANG class, 6-15
- NLS\_LANG constructor, 6-15
- NLS2Unicode
  - NLS class, 5-3
- NLSBufferSize
  - NLS class, 5-5
- NLSRuntimeException class
  - oracle.lang package, 5-6
  - oracle.owas.nls package, 6-26
- NLSRuntimeException constructor
  - oracle.lang package, 5-6
  - oracle.owas.nls package, 6-26
- NONE constant
  - IHAlign interface, 1-6
  - ITableRules interface, 1-10
  - IValign interface, 1-12
- NonUniformSizeException class, 7-43
- NonUniformSizeException constructor, 7-43
- NORMAL\_DOC constant
  - HtmlPage class, 1-125
- noWrap
  - TableDataCell class, 1-257
- NullSessionStateException class, 2-14
- NullSessionStateException constructor, 2-14

NullValueException class, 7-44  
NullValueException constructor, 7-44  
NumOptions  
    Select class, 1-221

## O

---

ObjectLookupFailureException class, 2-15  
ObjectLookupFailureException constructor, 2-15  
ObjectNotFoundException class, 2-17  
ObjectNotFoundException constructor, 2-17  
ObjectUpdateFailureException class, 2-19  
ObjectUpdateFailureException constructor, 2-19  
olive constant  
    Color class, 1-37  
Option class, 1-197  
Option constructor, 1-198  
ORACLE\_CHARSET constant  
    CharacterSetManager class, 6-13  
oracle.html package, B-1  
oracle.lang package, B-1  
oracle.OAS.Services.ICX package, B-1  
oracle.OAS.servlet.http package, B-1  
oracle.owas.nls package, B-1  
oracle.owas.wrb.WRB package, B-1  
oracle.plsql package, B-1  
oracle.rdbms package, B-1  
orange constant  
    Color class, 1-37  
ORDERED constant  
    List class, 1-183  
OrderedList class, 1-200  
OrderedList constructor, 1-201

## P

---

Paragraph class, 1-202  
Paragraph constant  
    SimpleItem class, 1-223  
Paragraph constructor, 1-202  
PARENT constant  
    ITarget interface, 1-11  
PasswordField class, 1-205  
PasswordField constructor, 1-206  
PBoolean class, 7-4

PBoolean constructor, 7-4  
PByteArray class, 7-8  
PDate class, 7-14  
PDate constructor, 7-14  
PDouble class, 7-19  
PDouble constructor, 7-19  
pink constant  
    Color class, 1-37  
PInteger class, 7-23  
PInteger constructor, 7-23  
plsql package, B-1  
PLSQLRuntimeErrorException class, 7-45  
PLSQLRuntimeErrorException constructor, 7-45  
positionOf  
    Style class, 1-231  
Preformat class, 1-207  
Preformat constructor, 1-208  
print  
    HtmlStream class, 1-135  
    HtmlStringBuffer class, 1-144  
    IHtmlItem interface, 1-7  
    IHtmlItemImpl class, 1-147  
printHeader  
    HtmlPage class, 1-129  
println  
    HtmlStream class, 1-136  
properties  
    StyleElement class, 1-235  
propertyAt  
    StyleElement class, 1-235  
PStringBuffer class, 7-27  
PStringBuffer constructor, 7-27  
putValue  
    HttpSession class, 2-8  
PValue class, 7-38

## R

---

Radio class, 1-209  
rdbms package, B-1  
red constant  
    Color class, 1-37  
removeElementAt  
    Style class, 1-229  
removeValue

- HttpSession class, 2-9, 2-10
- Reset class, 1-212
- Reset constructor, 1-212
- RHS constant
  - ITableFrame interface, 1-9
- RIGHT constant
  - IHAlign interface, 1-6
- rollback
  - Session class, 8-12
- ROWS constant
  - ITableRules interface, 1-10
- rowSpan
  - TableDataCell class, 1-257
- rules
  - Table class, 1-248
- run
  - WRBRunnable class, 3-3

## S

---

- Script class, 1-215
- Script constructor, 1-216
- Select class, 1-219
- Select constructor, 1-220
- SELF constant
  - ITarget interface, 1-11
- ServerException class, 8-15
- ServerException constructor, 8-15
- Session class, 8-3
- Session constructor
  - does not log in, 8-3
  - logs in, 8-4
- SessionDestroyFailureException class, 2-21
- SessionDestroyFailureException constructor, 2-21
- SessionException class, 2-23
- SessionException constructor, 2-23
- SessionLockedException class, 2-25
- SessionLockedException constructor, 2-25
- SessionLookupFailureException class, 2-27
- SessionLookupFailureException constructor, 2-27
- SessionNotFoundException class, 2-29
- SessionNotFoundException constructor, 2-29
- setAccept
  - FormElement class, 1-80
- setAcceptLanguage

- NLS\_LANG class, 6-21
- setActivatedLinkColor
  - HtmlBody class, 1-110
  - HtmlPage class, 1-130
- setAlign
  - Applet class, 1-22
  - FormElement class, 1-81
  - XObject class, 1-282
- setAltHtml
  - Applet class, 1-22
  - Frameset class, 1-93
- setAltText
  - Applet class, 1-23
- setAttr
  - Item class, 1-163
- setAuthInfo
  - ICXRequest class, 4-6
- setBackgroundColor
  - HtmlBody class, 1-111
  - HtmlPage class, 1-131
  - Table class, 1-243
  - TableCell class, 1-251
  - TableRow class, 1-265
- setBackgroundImage
  - HtmlBody class, 1-111
  - HtmlPage class, 1-131
  - Table class, 1-243
- SetBase
  - HtmlHead class, 1-121
- setBaseTarget
  - HtmlHead class, 1-122
- setBody
  - HtmlPage class, 1-129
- setBold
  - Item class, 1-164
- setBorder
  - Table class, 1-244
  - XObject class, 1-283
- setBorderColor
  - Table class, 1-244
  - TableCell class, 1-252
  - TableRow class, 1-265
- setCaption
  - Table class, 1-245
- setCellAt

- DynamicTable class, 1-64
- TableRow class, 1-264
- setCellPadding
  - Table class, 1-246
- setCellSpacing
  - Table class, 1-246
- setCenter
  - Item class, 1-164
- setCharacterSet
  - NLS\_LANG class, 6-22
- setCite
  - Item class, 1-165
- setClassID
  - XObject class, 1-283
- setCode
  - Applet class, 1-23
  - Item class, 1-165
  - Script class, 1-216
- setCodeBase, 1-284
- setCodebase
  - Applet class, 1-24
- setCodeType
  - XObject class, 1-284
- setCols
  - Frameset class, 1-94
  - TextArea class, 1-272
- setColSpan
  - TableDataCell class, 1-256
- setContent
  - ICXRequest class, 4-6
- setContentts
  - ICXRequest class, 4-7
- setCurrentCharSet
  - NLS class, 5-2
- setData
  - XObject class, 1-285
- setDeclare
  - XObject class, 1-285
- setDefaultCheckValue
  - FormElement class, 1-81
- setDefaultContent
  - TextArea class, 1-273
- setDefaultValue
  - FormElement class, 1-82
- setDefinition
  - Item class, 1-166
- setElementAt
  - Style class, 1-229
- setEmphasis
  - Item class, 1-166
- setFollowedLinkColor
  - HtmlBody class, 1-110
  - HtmlPage class, 1-130
- setFontBig
  - Item class, 1-167
- setFontColor
  - Item class, 1-167
- setFontSize
  - Item class, 1-168
- setFontSmall
  - Item class, 1-168
- setFontSubscript
  - Item class, 1-169
- setFontSuperscript
  - Item class, 1-169
- setForegroundColor
  - HtmlBody class, 1-111
  - HtmlPage class, 1-131
- setFrame
  - Table class, 1-247
- setHAlign
  - TableCell class, 1-251
  - TableRow class, 1-266
- setHead
  - HtmlPage class, 1-129
- setHeader
  - ICXRequest class, 4-7
- setHeaders
  - ICXRequest class, 4-8
- setHeading
  - Item class, 1-169
- setHeight
  - Applet class, 1-24
  - Embed class, 1-68
  - TableDataCell class, 1-256
  - XObject class, 1-286
- setHSpace
  - Applet class, 1-25
  - XObject class, 1-287
- setIAlign

- Table class, 1-247
- setItal
  - Item class, 1-170
- setItemAt
  - CompoundItem class, 1-46
  - Container class, 1-52
  - HtmlFile class, 1-116
  - HtmlPage class, 1-132
  - List class, 1-186
- setKeyboard
  - Item class, 1-170
- setLanguage
  - NLS\_LANG class, 6-23
- setLogStream
  - Session class, 8-13
- setMarginHeight
  - Frame class, 1-88
- setMarginWidth
  - Frame class, 1-88
- setMaxInactiveInterval
  - HttpSession class, 2-9
- setMaxLength
  - FormElement class, 1-82
- setMethod
  - ICXRequest class, 4-8
- setName
  - Anchor class, 1-17
  - Applet class, 1-25
  - FormElement class, 1-83
  - Frame class, 1-88
  - XObject class, 1-287
- setNoResize
  - Frame class, 1-89
- setNoWrap
  - TableDataCell class, 1-257
- setNull
  - PValue class, 7-41
- setProperties
  - Session class, 8-13
- setProperty
  - Session class, 8-9
- setPropertyAt
  - StyleElement class, 1-235
- setRows
  - Frameset class, 1-94
- TextArea class, 1-272
- setRowSpan
  - TableDataCell class, 1-257
- setRules
  - Table class, 1-248
- setSample
  - Item class, 1-171
- setScrolling
  - Frame class, 1-89
- setSessionProperty
  - Session class, 8-14
- setShapes
  - XObject class, 1-288
- setSize
  - FormElement class, 1-84
- setSrc
  - Embed class, 1-69
  - Frame class, 1-89
- setSrcURL
  - FormElement class, 1-84
- setStandby
  - XObject class, 1-288
- setStrike
  - Item class, 1-171
- setStrongEmphasis
  - Item class, 1-172
- setStyleElement
  - Item class, 1-172
- setTeletype
  - Item class, 1-173
- setTerritory
  - NLS\_LANG class, 6-23
- setTitle
  - HtmlHead class, 1-123
  - HtmlPage class, 1-132
- setUnderline
  - Item class, 1-174
- setUnfollowedLinkColor, 1-110
  - HtmlPage class, 1-130
- setUnicodeOutput
  - HtmlStream class, 1-137
- setUseMap
  - XObject class, 1-289
- setVAlign
  - TableCell class, 1-252

- TableRow class, 1-266
- setValue
  - NLS\_LANG class, 6-23
  - PBoolean class, 7-6
  - PByteArray class, 7-12
  - PDate class, 7-17
  - PDouble class, 7-21
  - PInteger class, 7-25
  - PStringBuffer class, 7-33
- setVariable
  - Item class, 1-174
- setVSpace
  - Applet class, 1-25
  - XObject class, 1-287
- setWalletInfo
  - ICXRequest class, 4-8
- setWidth
  - Applet class, 1-26
  - Embed class, 1-68
  - Table class, 1-248
  - TableCell class, 1-257
  - XObject class, 1-286
- setWRBContext
  - WRBRunnable class, 3-4
- silver constant
  - Color class, 1-37
- SimpleItem class, 1-223
- SimpleItem constructor, 1-224
- SINGLE\_QUOTE\_CHAR variable
  - HtmlStringBuffer class, 1-140
- size
  - CompoundItem class, 1-47
  - Container class, 1-53
  - List class, 1-186
  - Select class, 1-221
  - Style class, 1-231
  - StyleElement class, 1-236
- startTag
  - CheckBox class, 1-34
  - Frame class, 1-88
  - Frameset class, 1-93
  - Image class, 1-152
  - Radio class, 1-210
  - Reset class, 1-213
  - Table class, 1-247
- Str constant
  - IAlign interface, 1-6
  - ITableFrame interface, 1-9
  - ITableRules interface, 1-10
  - ITarget interface, 1-11
  - IValign interface, 1-12
- stringBufferValue
  - PStringBuffer class, 7-34
- stringToBytes
  - CharacterSet class, 6-12
- stringValue
  - PStringBuffer class, 7-35
- Style class, 1-227
- StyleElement class, 1-233
- StyleElement constructor, 1-234
- Submit class, 1-238
- Submit constructor, 1-239

---

## T

- Table class, 1-241
- Table constructor, 1-242
- TableCell class, 1-250
- TableCell constructor, 1-250
- TableCell class, 1-253
- TableCell constructor, 1-254
- TableHeaderCell class, 1-260
- TableHeaderCell constructor, 1-261
- TableRow class, 1-262
- TableRow constructor, 1-263
- TableRowCell class, 1-268
- TableRowCell constructor, 1-269
- teal constant
  - Color class, 1-37
- TextArea class, 1-271
- TextArea constructor, 1-272
- TextField class, 1-275
- TextField constructor, 1-276
- theStream
  - HtmlStream class, 1-138
- toCharArray
  - PStringBuffer class, 7-35
- toHTML
  - Anchor class, 1-18
  - Applet class, 1-26

- BaseFont class, 1-29
- CheckBox class, 1-34
- Comment class, 1-41
- CompoundItem class, 1-47
- Container class, 1-53
- DefinitionList class, 1-57
- DynamicTable class, 1-64
- Embed class, 1-69
- Font class, 1-73
- FormElement class, 1-85
- Frame class, 1-90
- Frameset class, 1-95
- Heading class, 1-101
- HeadLink class, 1-97
- Hidden class, 1-104
- HorizontalRule class, 1-106
- HtmlBody class, 1-112
- HtmlFile class, 1-117
- HtmlHead class, 1-123
- HtmlPage class, 1-133
- HtmlStringBuffer class, 1-145
- IHtmlItem interface, 1-8
- IHtmlItemImpl class, 1-147
- Image class, 1-152
- ImageMap class, 1-156
- ImageMapArea class, 1-158
- Item class, 1-175
- Link class, 1-181
- LinkBreak class, 1-178
- List class, 1-187
- ListItem class, 1-191
- MetaInfo class, 1-195
- Option class, 1-198
- Paragraph class, 1-203
- Radio class, 1-211
- Reset class, 1-213
- Script class, 1-217
- Select class, 1-222
- SimpleItem class, 1-224
- Style class, 1-232
- StyleElement class, 1-236
- Submit class, 1-239
- TableDataCell class, 1-258
- TableRow class, 1-266
- TableRowCell class, 1-269

- TextArea class, 1-274
- XObject class, 1-289
- TOP constant
  - ITarget interface, 1-11
  - IValign interface, 1-12
- toString
  - Anchor class, 1-19
  - Applet class, 1-27
  - BaseFont class, 1-30
  - Color class, 1-39
  - Comment class, 1-42
  - CompoundItem class, 1-48
  - Container class, 1-54
  - DynamicTable class, 1-65
  - Embed class, 1-70
  - Font class, 1-73
  - Form class, 1-77
  - Heading class, 1-102
  - HeadLink class, 1-98
  - HorizontalRule class, 1-107
  - HtmlBody class, 1-112
  - HtmlFile class, 1-117
  - HtmlHead class, 1-123
  - HtmlPage class, 1-133
  - HtmlStringBuffer class, 1-145
  - Image class, 1-152
  - ImageMap class, 1-156
  - ImageMapArea class, 1-159
  - Item class, 1-176
  - LineBreak class, 1-178
  - Link class, 1-182
  - List class, 1-187
  - ListItem class, 1-191
  - MetaInfo class, 1-196
  - NLS\_LANG class, 6-24
  - Paragraph class, 1-204
  - PBoolean class, 7-6
  - PByteArray class, 7-12
  - PDate class, 7-18
  - PDouble class, 7-22
  - PInteger class, 7-25
  - PStringBuffer class, 7-36
  - PValue class, 7-42
  - Script class, 1-217
  - Select class, 1-222

- SimpleItem class, 1-226
- Style class, 1-232
- StyleElement class, 1-237
- TableCell class, 1-258
- TableRow class, 1-267
- TableRowCell class, 1-270
- XObject class, 1-290
- TYPE\_CHECKBOX constant
  - FormElement class, 1-78
- TYPE\_FILE constant
  - FormElement class, 1-78
- TYPE\_HIDDEN constant
  - FormElement class, 1-78
- TYPE\_IMAGE constant
  - FormElement class, 1-78
- TYPE\_PASSWORD constant
  - FormElement class, 1-78
- TYPE\_RADIO constant
  - FormElement class, 1-78
- TYPE\_RESET constant
  - FormElement class, 1-78
- TYPE\_SUBMIT constant
  - FormElement class, 1-78
- TYPE\_TEXTFIELD constant
  - FormElement class, 1-78

## U

---

- Unicode2NLS
  - NLS class, 5-3
- unicodeOutput
  - HtmlStream class, 1-137
- UNORDERED constant
  - List class, 1-183
- UnOrderedList class, 1-277
- UnOrderedList constructor, 1-278
- useBuffer
  - PByteArray class, 7-13
  - PStringBuffer class, 7-36
- usingProxy
  - ICXResponse class, 4-14

## V

---

- valign

- TableCell class, 1-252
- variables
  - base
    - InvalidNameException class, 2-12
    - ObjectLookupFailureException class, 2-15
    - ObjectNotFoundException class, 2-17
    - ObjectUpdateFailureException class, 2-19
    - SessionDestroyFailureException class, 2-21
    - SessionException class, 2-23
    - SessionLockedException class, 2-25
    - SessionLookupFailureException class, 2-27
    - SessionNotFoundException class, 2-29
  - DOUBLE\_QUOTE\_CHAR
    - HtmlStringBuffer class, 1-140
  - NEWLINE\_CHAR
    - HtmlStringBuffer class, 1-140
  - SINGLE\_QUOTE\_CHAR
    - HtmlStringBuffer class, 1-140
  - VOLD constant
    - ITableFrame interface, 1-9
  - VSIDES constant
    - ITableFrame interface, 1-9

## W

---

- white constant
  - Color class, 1-37
- width
  - Table class, 1-248
  - TableCell class, 1-257
- WRB package, B-1
- WRBNotRunningException class, 3-5
- WRBNotRunningException constructor, 3-5
- WRBRunnable class, 3-2
- WRBRunnable constructor, 3-2
- WRBRuntimeException class, 3-6
- WRBRuntimeException constructor, 3-6
- write
  - HtmlStream class, 1-138

## X

---

- XObject class, 1-279
- XObject constructor, 1-280

## Y

---

yellow constant

Color class, 1-37

